ALFAMA CURSOS

alfamacursos.com.br

Alfama Cursos

Antônio Garcez
Fábio Garcez
Diretores Geral

Antônio Álvaro de Carvalho
Diretor Acadêmico

MATERIAL DIDÁTICO

Produção Técnica e Acadêmica

Marcela Menezes Flores
Coordenadora Geral

Patrícia Queiroz de Meneses
Coordenadora Pedagógica

José Alves Correia Neto
Renata Jacomo Viana
Autoria

Gabriella Caroline Teles Silva
Sabina Regina Conceição Santos
Revisão Textual

Rafael Rezende de Farias
Editoração

Todos os direitos reservados e protegidos pela Lei 9.610 de 19/02/98.
É proibida a reprodução total ou parcial, por quaisquer meios, sem autorização prévia, por escrito, da ALFAMA CURSOS.

alfamacursos.com.br

Arquitetura de Computadores

Apresentação do Curso

A disciplina de Arquitetura de Computadores apresentará ao aluno os conceitos básicos e fundamentais sobre como a sua principal ferramenta de trabalho, o computador, funciona do ponto de vista da execução de programas. Compreendendo o funcionamento interno dos computadores torna-se mais fácil descobrir as limitações, que podem ser encontradas em uma determinada família de processadores, para a solução de determinado tipo de problema computacional. Desse modo o aluno saberá escolher a arquitetura mais adequada para cada tipo de necessidade.

Apresentação do Professor

ROBERTO SILVA SANTOS Júnior é graduado em Desenvolvimento de Sistemas para Internet e pós-graduado em Gestão Tecnológica da Informação pela FANESE – SE. Atualmente é analista de suporte da Secretaria de Estado da Saúde de Sergipe, responsável pelo projeto de implantação do SINAVISA (Sistema de Informação Nacional de Vigilância Sanitária).

Componente Curricular

EMENTA

Introdução; Breve Histórico da Arquitetura de Computadores; Processadores; Arquitetura RISC X CISC; Paralelismo; Pipeline; Paralelismo de Processador; Memória Principal; Barramentos; Dispositivos de Entrada e Saída; Portas de Comunicação; Revisão.

OBJETIVOS

Instrumentalizar o aluno para que possa identificar e descrever o funcionamento de um sistema de computação e também como é realizada a integração entre o software e o hardware, bem como identificar as características da linguagem de máquina, as formas de endereçamento e o conjunto de instruções.

COMPETÊNCIAS

- Conhecer e entender as funcionalidades de um computador.
- Conhecer a evolução dos computadores.
- Identificar tipos de processadores e suas funcionalidades.
- Identificar os tipos de processamentos.
- Identificar os diferentes tipos de memória.
- Conhecer os diversos tipos de periféricos de entrada e saída.
- Identificar as portas de comunicação de um computador.

PÚBLICO-ALVO

Pessoas que estejam cursando ou tenham concluído o Ensino Médio maiores de 18 anos.

Índice

1 - INTRODUÇÃO 8
2 - BREVE HISTÓRICO DA ARQUITETURA DOS COMPUTADORES 9
2.1 - GERAÇÃO ZERO (1642 – 1945) 9
2.2 - PRIMEIRA GERAÇÃO (1945 – 1955) 10
2.3 - SEGUNDA GERAÇÃO (1955 – 1965) 12
2.4 - TERCEIRA GERAÇÃO (1965 – 1980) 13
2.5 - QUARTA GERAÇÃO (1980 –?) 14
3 - PROCESSADORES 17
4 - ARQUITETURA RISC X CISC 21
5 - PARALELISMO 22
6 - PIPELINE 23
7 - PARALELISMO DE PROCESSADOR 26
8 - MEMÓRIA PRINCIPAL 28
8.1 - MEMÓRIAS CACHE 29
8.2 - TIPOS DE ENCAPSULAMENTO DE MEMÓRIA 30
8.3 - SDR, DDR e DDR2 30
9 - BARRAMENTOS 33
9.1 - BARRAMENTO DO PROCESSADOR 34
9.2 - BARRAMENTO DE CACHE 34
9.3 - BARRAMENTO DA MEMÓRIA 34
9.4 - BARRAMENTO DE E/S 35
9.5 - BARRAMENTO ISA 35
9.6 - BARRAMENTO MCA 36
9.7 - BARRAMENTO EISA 36
9.8 - LOCAL BUS 36
9.9 - VESA LOCAL BUS 38
9.10 - BARRAMENTO PCI 38
9.11 - AGP 39
9.12 - PCI-EXPRESS 41
9.13 - BARRAMENTO PC-CARD (PCMCIA) 42
9.14 - USB 42
9.15 - FIREWIRE (IEEE 1394) 43
10 - DISPOSITIVOS DE ENTRADA E SAÍDA 44
11 - PORTAS DE COMUNICAÇÃO 46
11.1 - PORTAS SERIAIS 46
11.2 - USB 46
11.3 - FIREWIRE 47
12 - REVISÃO 48
12.1 - SISTEMA BINÁRIO 48
12.2 - OPERAÇÕES LÓGICAS COM BIT´S 48
12.3 - AND (E) 49
12.4 - OR (OU) 49
12.5 - XOR (OU exclusivo) 49
12.6 - NOT (NÃO) 50
12.7 - OPERAÇÕES LÓGICAS COM BYTES 50
12.8 - OPERAÇÕES MATEMÁTICAS COM BITS 50
12.8.1 - SOMA 50
12.8.2 - SUBTRAÇÃO 51
12.8.3 - MULTIPLICAÇÃO 51
12.8.4 - DIVISÃO 51
REFERÊNCIAS 53

Capítulo 1 - Introdução

1 - INTRODUÇÃO

O **computador** é uma máquina capaz de variados tipos de tratamento automático de informações ou processamento de dados. Um computador pode prover-se de inúmeros atributos, dentre eles armazenamento de dados, processamento de dados, cálculo em grande escala, desenho industrial, tratamento de imagens gráficas, realidade virtual, entretenimento e cultura.

No passado, o termo já foi aplicado a pessoas responsáveis por algum cálculo. Em geral, entende-se por computador um sistema físico que realiza algum tipo de computação. Existe ainda o conceito matemático rigoroso, utilizado na teoria da computação.

Assumiu-se que os computadores pessoais e laptops são ícones da Era da Informação e, isto é, o que muitas pessoas consideram como "computador". Entretanto, atualmente as formas mais comuns de computador em uso são os sistemas embarcados, pequenos dispositivos usados para controlar outros dispositivos, como robôs, câmeras digitais ou brinquedos.

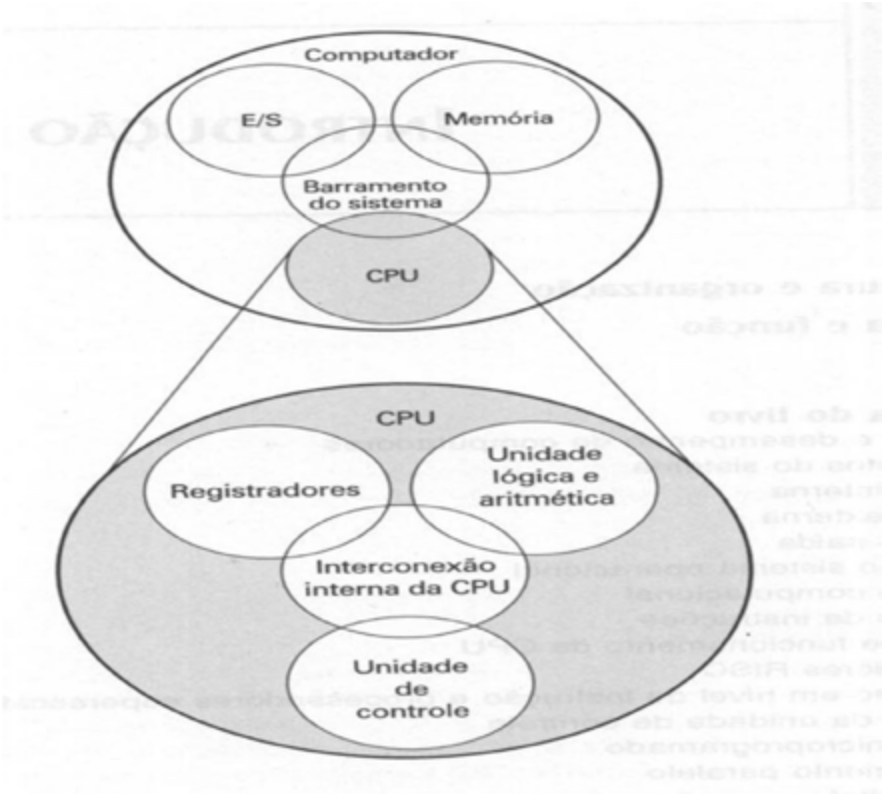


Figura 1

Os principais elementos de um sistema de computação são a unidade central de processamento (Central Processing Unit – CPU), a memória principal, o subsistema de E/S (entrada e saída) e os mecanismo de interconexão entre esses componentes. A CPU, por sua vez, consiste em uma unidade de controle, uma unidade lógica e aritmética (Aritmética and Logic Unit – ULA), registradores internos mecanismos de interconexão.

2 - BREVE HISTÓRICO DA ARQUITETURA DOS COMPUTADORES

2.1 - GERAÇÃO ZERO (1642 – 1945)

Em 1642, Blaise Pascal criou uma máquina para ajudar ao seu pai, que era coletor de impostos do governo francês. A máquina era totalmente mecânica, com várias engrenagens e alavancas e realizava somente operações de soma e subtração, mas foi responsável pelo marco zero da arquitetura dos computadores. A Linguagem Pascalina tem esse nome em homenagem a Blaise.

Figura 2 - Máquina de Pascal

Em 1694, um grande matemático alemão, o Barão Gottfried Wilhelm Von Leibnitz, construiu uma outra máquina que além de somar e subtrair, também realizava multiplicação e divisão. Fisicamente a máquina se parecia muito com a de Pascal.

Figura 3 - Máquina de Leibniz

Após 150 anos da construção da máquina de Leibniz, um professor da Universidade de Cambridge, chamado de Charles Babbage, inventor do velocímetro criou sua máquina diferencial. Esta máquina foi criada para construir a tabela de números utilizada pela navegação naval dos EUA. A máquina foi construída para executar somente um algoritmo, o do método das diferenças finitas utilizando polinômios. A característica mais interessante desta máquina era como ela apresentava seus dados calculados: era perfurado um prato de cobre, mídia esta que acabou sendo a precursora dos cartões perfurados e dos CD-ROM's.

Figura 4 - Máquina Diferencial de Babbage

Apesar de a máquina diferencial funcionar razoavelmente bem, Babbage queria construir uma máquina que executasse mais de um algoritmo e passou a trabalhar em um novo projeto que foi chamado de máquina analítica. A máquina analítica possuía quatro componentes: a memória, a unidade de computação, a unidade de entrada e a unidade de saída.

A grande vantagem da máquina analítica é que as instruções executadas por ela eram passadas através de cartões perfurados pela unidade de entrada; as instruções poderiam buscar valores na memória que seriam calculados na unidade de computação e depois seus resultados eram escritos também em cartões na unidade de saída.

Considerando que a máquina podia executar diversas instruções utilizando uma linguagem de montagem simples, Babbage necessitava de alguém que construísse programas para sua máquina, então, contratou uma jovem chamada Ada Augusta Lovelac, que vinha a ser filha do famoso poeta inglês Lorde Byron. Desta forma Ada foi a primeira pessoa na história a programar para um computador. Em sua homenagem, o Departamento de Defesa dos EUA batizou de ADA uma linguagem produzida sob seu patrocínio.

Apesar da boa intenção de Babbage, a máquina analítica não funcionou a contento. A máquina precisava de milhares de engrenagens e mecanismos de tamanha precisão que não era possível construir com a tecnologia utilizada naquele tempo. Suas ideias eram bastante avançadas para a época.

No final da década de 1930, um jovem chamado Howard Aiken executava a mão um grande número de cálculos necessários para a conclusão de sua tese de doutorado em Harvard. Após a conclusão de seu trabalho, Aiken achava que deveria existir algum mecanismo capaz de realizar seus cálculos de forma automática. Após algumas pesquisas, Aiken descobriu a máquina de Babbage e decidiu construí-la utilizando relés.

Sua primeira máquina, o MARK I, construída em Harvard foi terminada em 1944. Aiken tentou construir um sucessor de sua máquina, o MARK II, mas quando ela foi terminada, as máquinas a relés já estavam obsoletas, surgia à Era da Eletrônica com as válvulas.

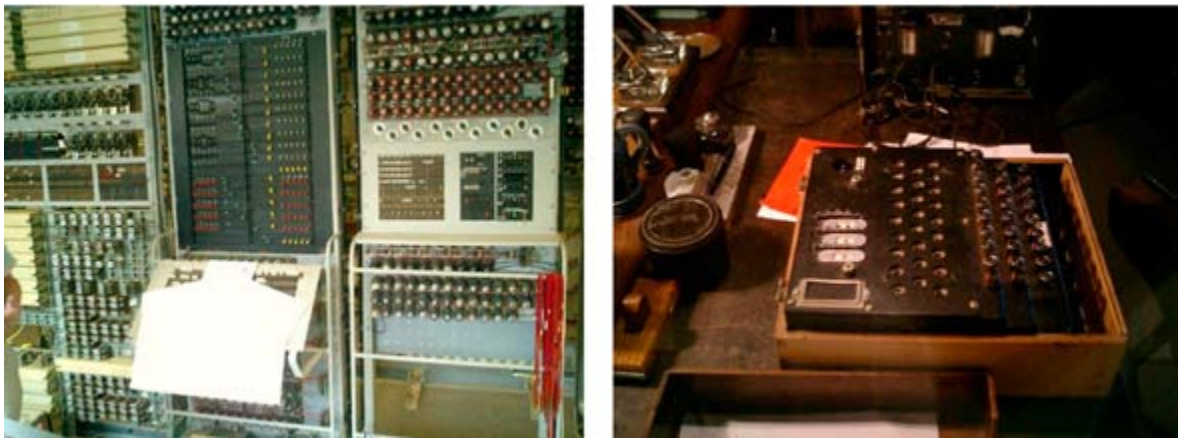
Figura 5 - MARK I

2.2 - PRIMEIRA GERAÇÃO (1945 – 1955)

Esta geração se caracteriza pela utilização de válvulas e começa com o COLOSSUS. Este computador, assim como as máquinas de Pascal e de Babbage foram vistas por você na disciplina de Introdução a Informática.

Figura 6 – Válvula

O COLOSSUS foi criado para decifrar códigos da máquina ENIGMA, utilizada pelos alemães na Segunda Guerra Mundial para passar mensagens cifradas para seus aliados. O projeto COLOSSUS foi desenvolvido na Inglaterra, em um laboratório altamente secreto. Alan Turing, famoso matemático britânico, participou deste projeto. O COLOSSUS tornou-se operacional em 1943, mas considerando que o governo britânico guardou como segredo militar, durante 30 anos todos os resultados da pesquisa, o projeto não se desenvolveu. Apesar disto o COLOSSUS é considerado por alguns autores como o primeiro computador eletrônico digital do mundo.



Figuras 7 - Réplica do COLOSSUS E 8 - Máquina ENIGMA

Aproximadamente nesta mesma época, John Mauchley, propôs um projeto ao exército americano para construção de um computador eletrônico para calcular tabelas de trajetórias balísticas. O projeto foi aceito e Mauchley, junto com um aluno de mestrado que ele orientava chamado J. Presper Eckert começou o projeto que foi chamado de ENIAC (Eletronic Numerical Integrator and Computer). Apesar de não se tornar operacional até 1946, após o fim da guerra foi considerado um sucesso e utilizado por nove anos. O ENIAC possuía 18.000 válvulas e 1.500 relés pesava 30 toneladas e consumia 140 quilowatts de energia elétrica.

Após o sucesso do ENIAC, Eckert e Mauchley tiveram o auxílio de John Von Newmann da Universidade de Princeton. Juntos eles trabalharam em um computador com armazenamento de programas chamado EDVAC. Um conflito entre eles fez com que o grupo se separasse. Eckert e Mauchley deixaram a Universidade da Pensilvânia e formaram sua própria empresa, a Eckert-Mauchley Computer Corporation, que após algumas fusões veio a se tornar a Unisys Corporation.

John Von Newmann decidiu construir sua própria versão do EDIVAC que foi chamada de máquina IAS. Algumas coisas que incomodavam Newmann eram o desconforto, a lentidão e a inflexibilidade de se programar os computadores por chaves e cabos. Ele imaginou que os programas poderiam ser representados de forma digital na memória do computador.
O projeto básico de Von Newmann é conhecido atualmente como "Máquina de Von Newmann".

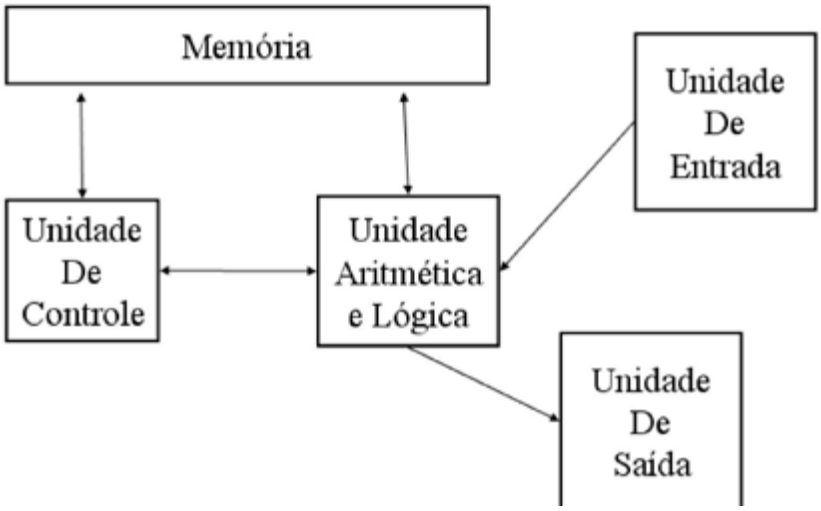


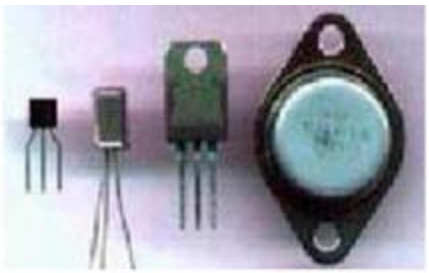
Figura 9 - Máquina de Von Newmann

Esta máquina tinha cinco componentes básicos: a memória, a unidade aritmética lógica, a unidade de controle, e os dispositivos de entrada e saída. As máquinas atuais são baseadas neste modelo.

Ainda nesta época, uma pequena empresa que produzia perfuradoras de cartões e máquinas mecânicas para separar cartões começou a produzir computadores, a IBM. Em 1953 a IBM produziu o computador chamado 701, disputando mercado com o UNIVAC, produzido pela empresa de Eckert e Mauchley. Ele foi o primeiro de uma séria de máquinas utilizadas em cálculos científicos com as quais a IBM dominou o mercado por uma década. Em 1958 a IBM produzia sua última máquina com válvulas, o 709.

2.3 - SEGUNDA GERAÇÃO (1955 – 1965)

Em 1947, no Bell Labs, John Bardeen, Walter Brattain e William Shockley inventaram o transistor, o que resultou em um Prêmio Nobel de Física em 1956. O transistor tem como funções principais amplificar e chavear sinais elétricos. As primeiras aplicações com os transistores foram em aparelhos auditivos e rádios transistorizados. Somente nos anos 60 os transistores começaram a substituir as válvulas nos computadores.



Figuras 10 - Transistores com vários encapsulamentos
e 11 - Transistor moderno de alta potência

O primeiro computador transistorizado foi construído no MIT e chamava-se TX-0 (Transistorized Experimental Computer 0) que funcionava a 16 bits. Os engenheiros que trabalharam no projeto do TX-0 fundaram a DEC que começou a produzir o PDP-1, rivalizando com a principal construtora de computadores na época, a IBM.

O PDP-1 possuía metade do desempenho do IBM 7090, máquina produzida pela IBM na época, mas custava cerca de 120.000 dólares contra milhares de dólares da 7090. A DEC vendeu dezenas destas máquinas iniciando a era dos minicomputadores.

Nesta época a IBM evoluiu o 7090 para o 7094, bastante utilizado para o mercado científico e começou a produzir o 1401 para aplicações comerciais. O 1401 era uma máquina com menor processamento que o 7094, mas servia muito bem para o que ela se destinava.

Em 1964, uma empresa chamada Control Data Corporation (CDC) lançou o 6600, máquina superior a 7094, computador mais rápido até então. O projetista do 6600 foi Seymour Cray que dedicou sua vida a produzir máquinas cada vez mais rápidas, conhecidas como supercomputadores. Deste podemos citar o próprio 6600, o 7600 e o Cray-1.

2.4 - TERCEIRA GERAÇÃO (1965 – 1980)

Em 1958, Robert Noyce desenvolveu um processo de integrar circuitos eletrônicos em substratos de silício, conseguindo colocar dezenas de transistores em um mesmo chip.

Figura 12 - Circuito Integrado

Com este componente foi possível construir máquinas mais rápidas, menores e mais baratas do que as máquinas construídas com transistores.

Em 1964, a IBM enfrentava um grande problema com suas principais máquinas que estavam no mercado: 7094 e 1401. As duas possuíam arquiteturas diferentes e consequentemente instruções diferentes, ou seja, os programas construídos para uma não rodavam na outra. Em função disto muitos dos clientes corporativos da IBM precisavam manter duas equipes de programação diferentes, gerando um alto custo para manter as duas máquinas.

Então a IBM decidiu realizar algumas mudanças na forma de construir seus novos computadores. Introduziu uma única linha de produto, o System/360, produzido com circuitos integrados. O System/360 não era somente um computador, mais seis com mesma linguagem de montagem, mas com poder de processamento e tamanhos variáveis. Com isso as empresas poderiam escolher qual máquina se adequaria ao seu tamanho, e caso fosse necessário realizar uma outra compra no futuro, não precisaria reescrever seus programas novamente.

O modelo 30 seria o equivalente ao IBM 1401 enquanto que o modelo 75 equivaleria ao 7094. Também como inovação, a família System/360 trouxe a multiprogramação, no qual o sistema mantinha diversos programas na memória ao mesmo tempo. Enquanto um programa estivesse esperando a conclusão de operações de entrada/saída, outro estaria processando.

Além da multiprogramação, as máquinas da família System/360 foram as primeiras a poder emular outros computadores. O menor de seus modelos podia perfeitamente simular o 1401 enquanto que os modelos maiores emulavam o 7094. Alguns dos modelos 360 rodavam os programas do 1401 mais rápidos que o próprio 1401. Desta forma os clientes antigos não precisavam reescrever seus programas, pois eles conseguiriam rodar nas novas máquinas.

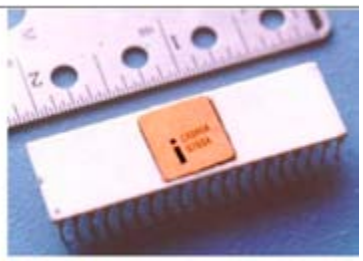
Figuras 13 - System/360 modelo 40 e 14 - Representação de um System/360

2.5 - QUARTA GERAÇÃO (1980 –?)

Na década de 80 uma nova tecnologia conseguiu colocar dezenas de milhares de transistores em um único chip. Esta tecnologia é conhecida como VLSI (Very Large Scale Integratrion) e continua a ser utilizada principalmente na fabricação de chips, pois hoje se pode colocar milhões de transistores em uma única pastilha de silício.

Esta tecnologia levou a indústria a criar máquinas menores e mais rápidas. Por volta de 1980 os computadores ficaram tão baratos que pessoas físicas começaram a comprá-los e utilizá-los em casa, resultando no início de era dos computadores pessoais.

Os primeiros modelos de computadores pessoais foram vendidos em forma de kits a serem montados pelos usuários, nesta época estudantes de eletrônica. Estes kits possuíam como processador o Intel 8080. Nenhum software vinha no kit, se fosse necessário algum, o próprio usuário teria que escrevê-lo. Com isso Gary Kildall escreveu um sistema operacional chamado de CP/M que foi bastante utilizado nas máquinas que possuíam chips Intel 8080.



Figuras 15 - Computador SOL - 8080 e 16 - Intel 8080

O Apple e o Apple II foram dois outros computadores pessoais que foram lançados também no início da década de 80. Foram projetados por Steve Jobs e Steve Wozniak que fundaram a Apple. Mais uma empresa que entrou no mercado de computadores pessoais. A história de Steve Jobs é interessante, conhecido por ser criador do iPod e fundador da Pixar, comprada recentemente pela Disney.

Figura 17 - Apple II

A IBM decidiu então entrar no mercado de computadores pessoais e solicitou que um dos seus executivos, Philip Estridge construísse um computador pessoal. Em 1981 a IBM produziu o Personal Computer (PC IBM) tornando-se o computador mais vendido da história.

Figura 18 - PC IBM

A IBM, ao contrário de todos os seus outros projetos que eram guardados a sete chaves, publicou um livro com todo o projeto do PC, inclusive o desenho dos circuitos. A ideia era com que outras empresas fabricassem placas compatíveis com o projeto para popularizar o PC. Mas o que decorreu disto foi que várias outras empresas começaram a fabricar clones dos PC's e vendê-los mais baratos.

A versão inicial do PC vinha com o sistema operacional MS-DOS fornecido pela recém-criada Microsoft. Enquanto a Intel tentava produzir chips mais velozes, a IBM e a Microsoft trabalhavam em um novo sistema operacional com interface gráfica, o OS/2, semelhante ao utilizado no Apple Macintosh. Ao mesmo tempo a Microsoft estava produzindo um outro sistema operacional, o Windows, que iria rodar em cima do DOS. Desta forma o OS/2 não vingou, a IBM brigou com a Microsoft e, o Windows virou um dos sucessos mais retumbantes de toda a história dos computadores.

Em meados da década de 80 começaram a surgir novos computadores utilizando tecnologia RISC, que veremos mais adiante. Nos anos 90, os fabricantes de chips começaram a desenvolver os processadores superescalares que podiam executar mais de uma instrução ao mesmo tempo.

Hoje, já temos processadores de vários núcleos e outros que funcionam a laser, além dos processadores quânticos, porém isto será visto com mais detalhes quando falarmos sobre processadores mais adiante.

Recordando

Vimos à história da arquitetura de computadores e a evolução para o computador moderno. Os principais pontos abordados foram:

- As máquinas criadas por Babbage.
- Primeiras máquinas com programas.
- Evolução dos componentes.
- Criação do sistema operacional.
- A criação de grandes empresas de informática.

3 - PROCESSADORES

Segundo Tanenbaum, o processador equivale ao cérebro do computador. Sua função é executar os programas armazenados na memória principal, buscando cada uma das instruções de um programa, examinando-as e executando-as uma a uma.

O processador possui uma estrutura como descrita na figura abaixo:

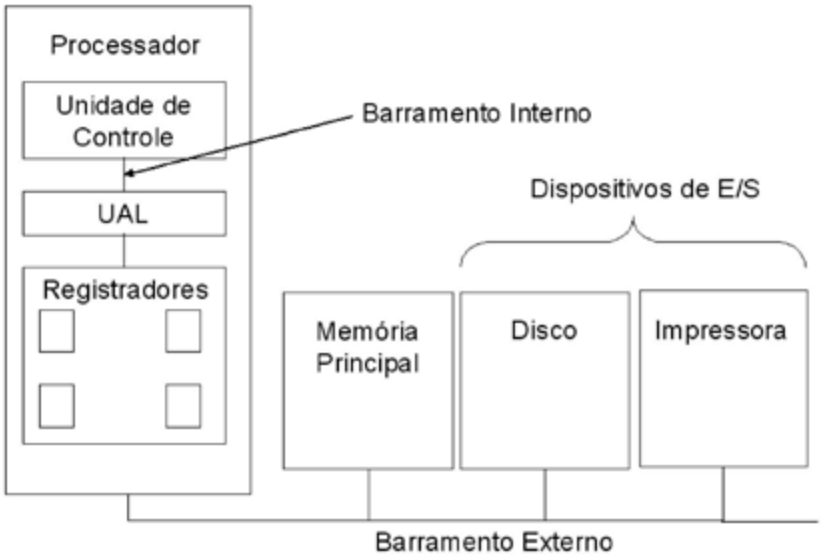


Figura 19 - Estrutura de um processador muito simples

Note que ele é composto por algumas partes:

O **barramento interno** é um conjunto de fios paralelos que possibilita a comunicação dos componentes presentes no processador. É o barramento interno que permite a transmissão de dados (transporte de informações), endereços (que identifica para onde a informação está sendo enviada) e sinais de controle (que descreve os aspectos sobre como a informação está sendo transmitida).

O **processador** também é composto por uma unidade de controle que é responsável pela busca da informação na unidade de memória principal e pela determinação do tipo de cada instrução.

Os **registradores** representam a memória do processador e se caracterizam por serem pequenos, rápidos e separados da memória principal do sistema. Esta memória é utilizada para armazenar resultados temporários e certas informações de controle. Por exemplo, neste conjunto de registradores, dois são utilizados pela unidade de controle: o **Program Counter** (PC) e o **Registrador de Instruções** (IR).

O registrador PC aponta para a próxima instrução que está na memória principal e que o processador deve executar. O registrador IR armazena a instrução que está sendo executada.

A Unidade Aritmética Lógica (UAL) realiza cálculos com os valores que estão presentes nos registradores. Estas operações são binárias ou unárias como, por exemplo, **a soma, e, não, ou, e multiplicação.**

Internamente estes três componentes do processador conversam entre si através do barramento interno, mas o processador precisa se comunicar com os outros periféricos da máquina, principalmente a memória. Para isso o processador utiliza um conjunto de fios semelhantes aos que possuem internamente. Este conjunto de fios que comunica o processador com os outros componentes de um computador é o barramento externo. O barramento externo possui uma velocidade menor que o barramento externo, ou seja, qualquer instrução que necessita de dados provenientes da memória principal, por ter que buscar o dado utilizando o barramento externo, será executada de forma mais lenta do que uma instrução que já possui seus dados nos registradores.

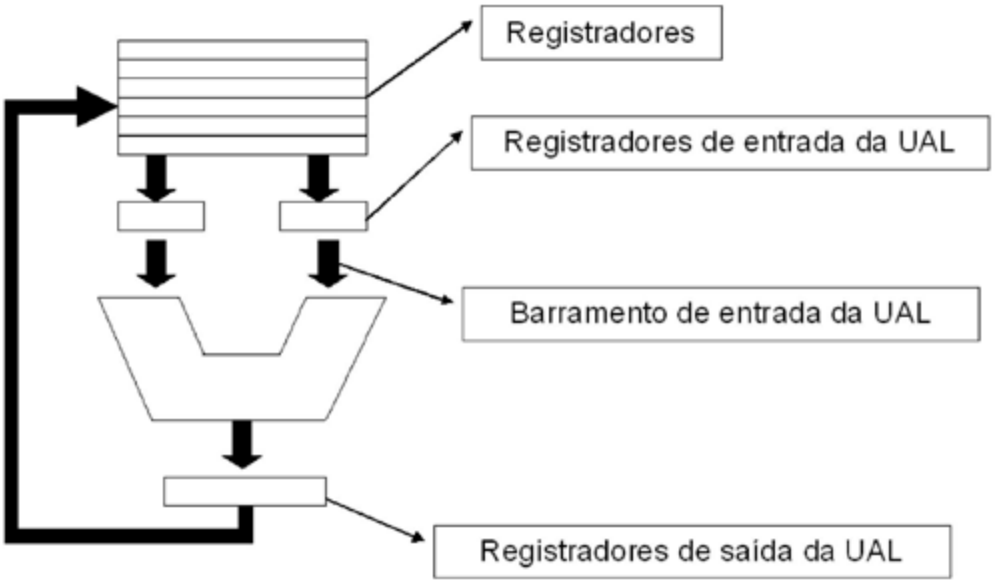


Figura 20 - Caminho de dados

A figura acima representa o caminho de dados que é por onde o dado passa para ser calculado. Verifique que antes de chegar a UAL, os dados devem estar em dois registradores de entrada. Após estes registradores serem preenchidos, a UAL realiza o cálculo e armazena no registrador de saída. Se a instrução for do tipo unária, somente um registrador de entrada é carregado.

O conjunto de instruções é a coleção de instruções que um processador pode executar. O conjunto de instruções para cada tipo de processador é diferente um do outro. As diferenças estão principalmente no tamanho das instruções, no tipo de operação permitida, no tipo de operandos que são utilizados e no tipo de resultado que é gerado.

As instruções podem ser classificadas, em sua maioria, em duas categorias: instruções registrador-registrador e instruções registrador-memória.

As instruções registrador-memória se caracterizam por necessitar de dados que estão na memória principal do computador. Como vimos anteriormente, o barramento externo é mais lento que o barramento interno. Neste caso estes tipos de instruções são mais lentas para serem executadas pelo processador.

As instruções registrador-registrador utilizam como operandos para cálculo na UAL valores que já estão nos registradores, não necessitando utilizar o barramento externo. Instruções deste tipo são mais rápidas em sua execução.

Vamos imaginar que o processador irá executar uma instrução de um pequeno programa. Esta instrução, que está na memória principal representa a soma de dois valores inteiros: **ADD AX, 5 ----->** Instrução que soma 5 ao conteúdo do registrador AX. Para execução desta instrução, o processador necessitaria realizar os seguintes passos:

1 - Buscar a instrução na memória principal e armazená-la no registrador IR.
2 - Atualizar o registrador PC.
3 - Determinar o tipo de instrução (se é registrador-registrador ou registrador-memória).
4 - Caso seja uma instrução registrador-memória, determinar o endereço de memória onde o valor necessário para execução da instrução está armazenado.
5 - Buscar o valor na memória principal e armazenar em um registrador
6 - Executar a instrução e armazenar o resultado.
7 - Retornar ao primeiro passo.

Esta sequência de passos é conhecida como **busca-decodificação-execução**, onde decodificação é a preparação da instrução para ser executada, determinação do seu tipo e busca dos valores necessários na memória principal.

O fabricante do processador precisa definir algumas coisas antes de produzir um chip. Primeiro definir qual a linguagem será utilizada e quais instruções seu processador irá executar (conjunto de instruções). Após isso se deve definir como o processador executará as instruções passadas para ele.

Os processadores sempre executarão uma linha de programa por vez, seguindo o que aprendemos na interpretação. Note que as instruções neste nível também serão traduzidas. Os programas são escritos utilizando o conjunto de instruções definido pelo fabricante e o processador deve traduzir estas instruções para seus componentes eletrônicos.

Esta tradução pode ser realizada de duas formas: por software ou por hardware. Na verdade tudo deve ser traduzido para o hardware, mas podemos ter um pequeno programa dentro do processador que irá realizar mais uma tradução, só que internamente.

Vamos entender melhor. Para facilitar o trabalho dos programadores, os fabricantes tendem a definir um conjunto de instruções bastante amplo, desta forma o programador possui uma variedade maior de operações para construírem seus programas, tornando a programação mais fácil. Mas com um grande conjunto de instruções, o fabricante deve produzir um hardware mais complexo para entender as operações passadas pelos programadores. Ora, se o hardware é mais complexo, então ele vai ser mais caro e ocupar um espaço maior, já que precisa de um conjunto maior de componentes eletrônicos.

Desta forma, os fabricantes de processadores decidiram colocar um pequeno programa dentro do processador que é capaz de traduzir para instruções mais simples as operações passadas pelo programador. Com isso, o hardware que executa as instruções de fato, se torna menos complexo e mais barato e o número de instruções não diminui para o programador deixando todo mundo contente.

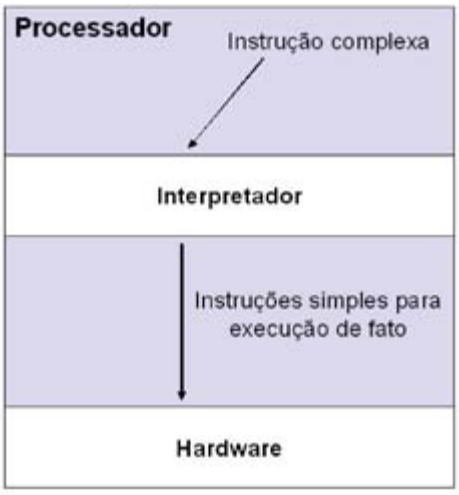


Figura 21 - Esquema de execução de um processador com interpretador

Além disto, as vantagens de utilizar um interpretador são várias:

- É mais fácil corrigir erros de implementação em um programa do que em componentes eletrônicos.
- É mais fácil incorporar novas instruções em um programa.
- O desenvolvimento, realização de testes e geração de documentação de um programa lógico é mais simplificado.

Mas, a utilização de um interpretador tem como principal problema o detrimento do desempenho do processador, pois se faz necessário mais uma camada (o interpretador) para a execução das instruções.

Para melhorar o desempenho do processador, um componente eletrônico foi fundamental, a memória de controle. As memórias de controle são memórias Read-only muito rápidas e capazes de armazenar o interpretador. Com este componente, a performance de execução das instruções por interpretador não era tão pior do que um processador executando as instruções diretamente no hardware. Note que mesmo com as memórias de controle, um processador que executa instruções diretamente no hardware era mais veloz, mas o preço do processador interpretado e sua performance menor, mas atendendo as expectativas fez com que os processadores com interpretador tivessem um grande sucesso por um tempo.

4 - ARQUITETURA RISC X CISC

Nos últimos anos da década de 70, a técnica dominante para projeto de processadores era a utilização de interpretadores com instruções muito complexas. Mas um grupo de pesquisa da IBM, liderado por John Cocke pretendia seguir contra esta corrente. O grupo desenvolveu um protótipo de minicomputador chamado 801 que executava as instruções diretamente no hardware sem interpretador. Claro que o conjunto de instruções deste processador era bem pequeno. Apesar da IBM nunca ter comercializado esta máquina vazaram informações obre a pesquisa e vários outros grupos começaram a trabalhar nestes conceitos.

Em 1980, um grupo liderado por David Patterson e Carlo Séquin, em Berkleye, desenvolveu chips processadores que não utilizavam interpretação. Entenda que nesta época a velocidade das memórias RAM davam um grande salto de velocidade, tornando mais rápidas instruções vindas diretamente da memória.

Para designar este novo conceito, a equipe de Patterson usou o termo RISC (*Reduced Instruction Set Computer*) e chamaram o primeiro chip construído seguindo esta filosofia de RISC I que foi logo seguido do RISC II, produzido pela mesma equipe.

Em 1981, John Hennessy, pesquisador de Stanford, projetou e fabricou um chip diferente, também executando instruções diretamente no hardware, que ele chamou de MIPS. Este projeto se tornou um sucesso comercial e geraram produtos famosos como o chip SPARC e o chip MIPS.

Na época em que estes processadores estavam sendo fabricados, o que chamava bastante a atenção era o seu número de instruções bastante reduzido, em torno de 50, que era bastante menor que as 200 ou 300 instruções dos processadores que dominavam o mercado na época. A arquitetura utilizando interpretador e um conjunto de instruções maior dá-se o nome de CISC (*Complex Instruction Set Computer*).

Começou então uma "guerra" entre os defensores do CISC e os usuários do RISC. Com o aumento da velocidade da RAM, as máquinas CISC estavam perdendo bastante terreno em relação à performance. As máquinas fabricadas com arquitetura RISC, como o chip Alpha da DEC, tinham tudo para ser um enorme sucesso de vendas. Mas não obteve o sucesso esperado. Um dos problemas da nova arquitetura era compatibilidade. Para se usar a nova tecnologia, as empresas tinham que trocar o parque antigo, baseado em tecnologia Intel (CISC) o que representava jogar no lixo bilhões de dólares.

Outro fator foi que a Intel não queria perder seu mercado e começou a investir em pesquisas para melhorar a performance de seus chips sem abrir mão do conjunto complexo de instruções. A partir do 486, a Intel começou a utilizar um núcleo RISC dentro de seus processadores CISC. Com isso os processadores só conseguiam executar as instruções mais simples (a maioria das instruções) diretamente no hardware, enquanto que as mais complexas ainda eram interpretadas. Apesar de não ter a performance do RISC puro, esta tecnologia mantém a compatibilidade e ganhou uma relativa performance em relação aos chips antigos. Estes processadores são chamados de híbridos por comportar as duas tecnologias. É esta tecnologia que hoje domina o mercado.

5 - PARALELISMO

As pesquisas relacionadas a processadores têm sempre o desafio de construir novos chips que sejam mais baratos e que executem as instruções cada vez mais rápido. Mas existe um limite tecnológico para isso, apesar dos fabricantes sempre nos surpreenderem com novas formas de construção de processadores. Como consequência desta dificuldade, a grande maioria dos arquitetos de computadores voltou-se para o estudo da execução de mais de uma operação ao mesmo tempo, chamado de paralelismo.

O paralelismo se manifesta de duas maneiras: no nível de instruções e no nível do processador. O primeiro caso é a execução por parte do processador de mais instruções por segundo enquanto que na segunda opção temos vários processadores trabalhando juntos para a solução de um mesmo problema.

A computação paralela é uma forma de computação em que vários cálculos são realizados simultaneamente, operando sob o princípio de que grandes problemas geralmente podem ser divididos em problemas menores, que então são resolvidos concorrentemente (em paralelo). Existem diferentes formas de computação paralela: em bit, instrução, de dado ou de tarefa. A técnica de paralelismo já é empregada por vários anos, principalmente na computação de alto desempenho, mas recentemente o interesse no tema cresceu devido às limitações físicas que previnem o aumento de frequência de processamento. Com o aumento da preocupação do consumo de energia dos computadores, a computação paralela se tornou o paradigma dominante nas arquiteturas de computadores sob forma de processadores multinúcleo.

Computadores paralelos podem ser classificados de acordo com o nível em que o hardware suporta paralelismo. Computadores com multinúcleos ou multiprocesadores possuem múltiplos elementos de processamento em somente uma máquina, enquanto clusters, MPP e grades usam múltiplos computadores para trabalhar em uma única tarefa. Arquiteturas paralelas especializadas às vezes são usadas junto com processadores tradicionais, para acelerar tarefas específicas.

Programas de computador paralelos são mais difíceis de programar que sequenciais, pois a concorrência introduz diversas novas classes de defeitos potenciais, como a condição de corrida. A comunicação e a sincronização entre diferentes subtarefas é tipicamente uma das maiores barreiras para atingir grande desempenho em programas paralelos. O aumento da velocidade por resultado de paralelismo é dado pela Lei de Amdahl.

6 - PIPELINE

Há bastante tempo sabe-se que o maior problema para a velocidade de execução de instruções é o acesso à memória. Para minimizar este problema, as máquinas começaram a ser planejadas com a capacidade de buscar antecipadamente instruções na memória, de maneira que elas já estejam no processador quando chegar o momento de executá-las. Estas instruções são armazenadas em um conjunto de registradores que são chamados de buffer de pré-busca, que mais tarde evoluiu para as memórias caches que discutiremos mais adiante.

Além do conceito de pré-busca foram elaboradas formas de executar as instruções em várias partes. Este conceito foi chamado de **Pipeline**.

O pipeline funciona dividindo os passos para execução de uma instrução em vários hardwares, dentro do processador, específicos para cada operação. A execução destas partes é tratada em paralelo. Veja o esquema abaixo:

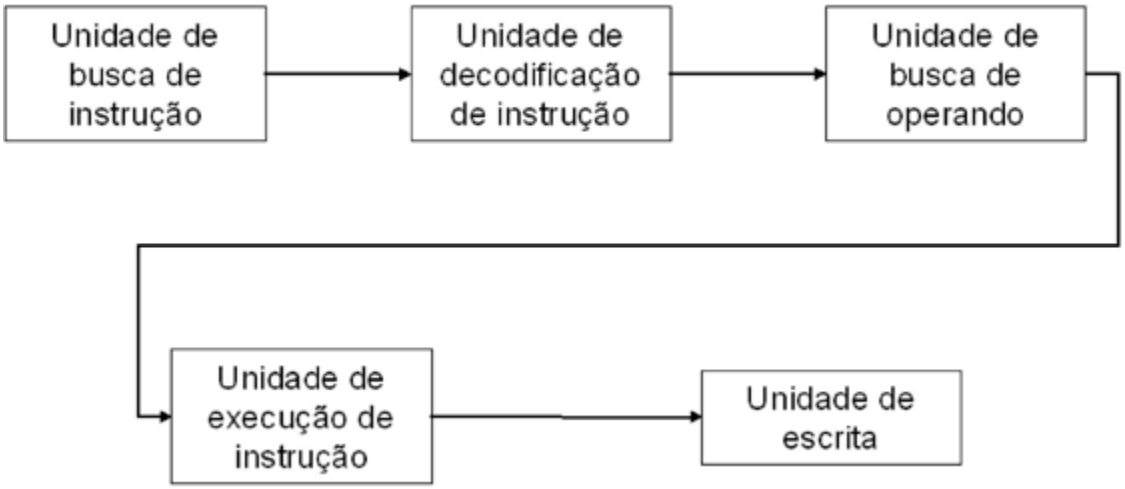

Unidade de busca de instrução
Unidade de decodificação de instrução
Unidade de busca de operando
Unidade de execução de instrução
Unidade de escrita

Figura 22 - Pipeline de cinco estágios

A figura representa um Pipeline de cinco estágios. Note que cada unidade do Pipeline representa uma das etapas de execução de uma instrução. Só que no pipeline, as etapas são executadas em conjunto, desta forma em um determinando momento teremos cinco instruções sendo processadas ao mesmo tempo. Vamos entender melhor.

Imagine que temos um conjunto de instruções para serem executadas no processador. Cada instrução irá passar por todas as etapas. Suponha que no primeiro ciclo de clock, a primeira instrução esteja no Estágio 1 (S1), ou seja, a instrução está sendo transportada da memória principal para o registrador IR (lembra dele?). No segundo ciclo de clock a instrução 1 irá passar para o segundo estágio, a decodificação da instrução (determinação do tipo de instrução) e neste mesmo ciclo a segunda instrução já pode ser transportada para os registradores. No terceiro clock, a primeira instrução já estará no terceiro estágio, enquanto que a segunda instrução passa para o segundo estágio, e a terceira instrução inicia no primeiro estágio. Veja a figura abaixo:

E1	I1	I2	I3	I4	I5
E2		I1	I2	I3	I4
E3			I1	I2	I3
E4				I1	I2
E5					I1

Figura 23 - Estágios de um Pipeline

Verifique que as instruções vão mudando de estágio com o passar do tempo. Em um determinado espaço de tempo, teremos cinco instruções sendo processadas ao mesmo tempo. Verifique o termo "processadas" que é diferente de "executadas", pois a instrução só será realmente executada no estágio 4.

Significa dizer que temos um ganho de cinco no processamento de instruções dentro do chip do processador. Mas na verdade o tempo na prática não é realmente este, pois uma instrução pode demorar mais de um ciclo sendo executada (estágio 4) atrasando assim as outras instruções. Mesmo com este tipo de ocorrência, experiências em laboratórios aprovaram um alto ganho de performance nos processadores que utilizam a tecnologia de Pipeline.

Com o tempo o pipeline foi evoluindo e chegamos a uma técnica chamada de processadores superescalares. Nestes processadores o Pipeline é aumentado no estágio quatro, **que pode se o mais demorado.**

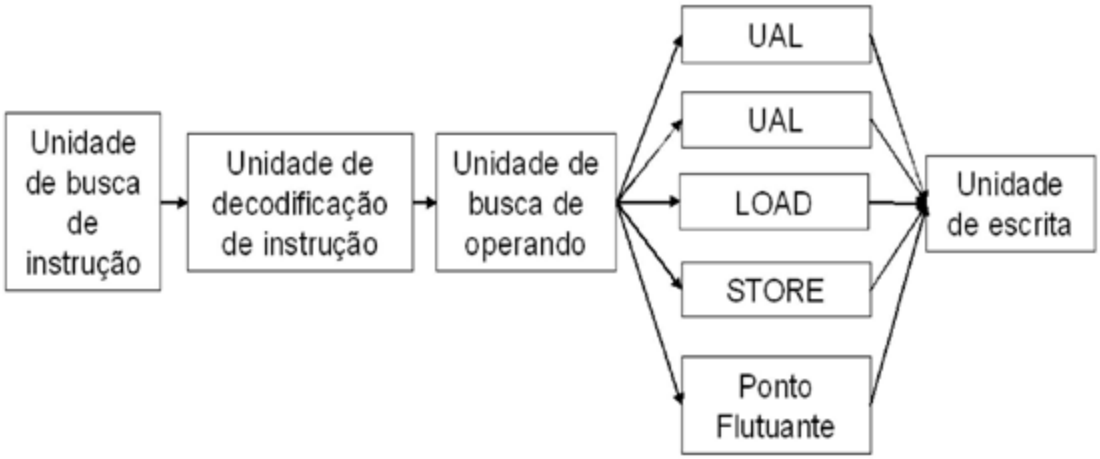


Figura 24 - Pipeline superescalar

O ganho de performance de um processador utilizando pipeline superescalar é muito alto. A Intel produz o Pentium 4 utilizando arquitetura superescalar. O P4 possui 6 unidades de execução para números inteiros e duas para números de ponto flutuante (números reais), além de ter um circuito de Branch Prediction que permite tentar adivinhar as tomadas de decisões de cada instrução. Esta técnica é bastante importante por tentar não atrasar o processamento das instruções. Imaginem que uma instrução pode depender de uma outra, como por exemplo utilizar a mesma região de memória para realizar seus cálculos. Estas instruções não podem ser executadas ao mesmo tempo, pois pode causar falha no resultado

final. Este mecanismo tenta adivinhar de forma precisa este tipo de acontecimento e toma as decisões necessárias para não atrapalhar a performance e não influenciar no resultado final da operação.

A Intel acaba de anunciar testes em laboratório sobre uma nova tecnologia de fabricação de processadores. Os processadores a laser:

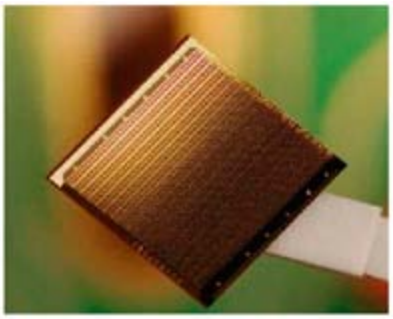


Figura 25 - Processador a laser

Esta nova tecnologia permite que a troca de informações interna ao processador (barramento interno) seja feita utilizando laser, passando por fios de fibra óptica. Esta nova tecnologia, segundo a Intel, aumentará em 100 vezes a capacidade de processamento do chip e será comercializada até o final desta década (2010).

7 - PARALELISMO DE PROCESSADOR

Mesmo com a evolução dos processadores, algumas operações necessitam de um poder de processamento que somente um processador não consegue realizar. Nestes casos é comum a utilização de máquinas com mais de um processador ou um conjunto de máquinas que funcionam como uma só, dividindo processamento. Este último caso nós chamamos de ***cluster.***

Máquinas multiprocessadas (mais de um processador) são bastante comuns hoje em dia, principalmente em máquinas que possuem perfil de servidores, como os computadores servidores de banco de dados, servidores de aplicações, servidores de arquivos, etc.

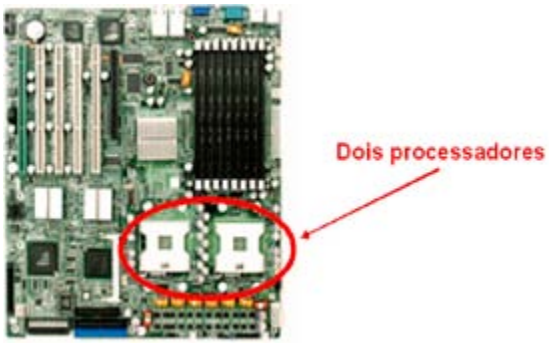


Figura 26 – Placa-mãe para dois processadores

Mesmo assim existem diversas aplicações que uma máquina multiprocessada não consegue realizar processamento suficiente. Nestes casos pode-se configurar mais de uma máquina dividindo os mesmos processos. Este conjunto de computadores é chamado de cluster.

Casos famosos como o cluster Columbia e o site Google requer bastante processamento para aplicações específicas.

O Columbia, supercomputador residente na NASA, possui 10.240 processadores em cluster, custou 200 milhões de dólares e realiza simulações de testes de segurança, o que evita a utilização de dispendiosos e perigosos equipamentos para tais fins. Os processadores utilizados no cluster possuem tecnologia Intel Itenium 2.

Figura 27 – Columbia

O Google, por outro lado, precisa de bastante processamento para indexação e busca dos sites cadastrados em sua base de dados. Segundo Patterson, o Google já indexou até hoje mais de 3 bilhões de páginas e, para realizar aproximadamente 1.000 consultas por segundo, dispõe de 6.000 processadores em cluster. Observe que não são seis mil máquinas, mas um conjunto de máquinas multiprocessadas em cluster que representam juntas este total de processadores.

Recordando

Neste capítulo compreendemos a execução de instruções em um processador e a evolução desta arquitetura. Os principais pontos abordados foram:

- Estrutura interna do processador.
- Máquinas CISC e RISC.
- Pipeline e superescalar.
- Cluster de computadores.

8 - MEMÓRIA PRINCIPAL

A função do processador é executar as funções armazenadas na memória principal. Todas as informações que o processador precisa se não estiverem em seus registradores devem ser buscadas na memória principal.

É comum encontrar em algumas literaturas que um programa, para ser executado deve estar na memória principal. Realmente é necessário que as instruções estejam na memória principal já que o processador não irá buscar nenhuma instrução que não seja em seus registradores ou na memória.

A memória principal é volátil, ou seja, não armazena valores quando está sem energia. Quando o computador é desligado, a memória principal perde todos os seus dados. Mais adiante falaremos das memórias secundárias que guarda valores mesmo sem energia, como os discos rígidos e CD-ROM.

A memória é dividida em células utilizadas para armazenar valores. Cada célula possui uma identificação que chamamos de endereço de memória. No esquema abaixo podemos visualizar como a memória é dividida.

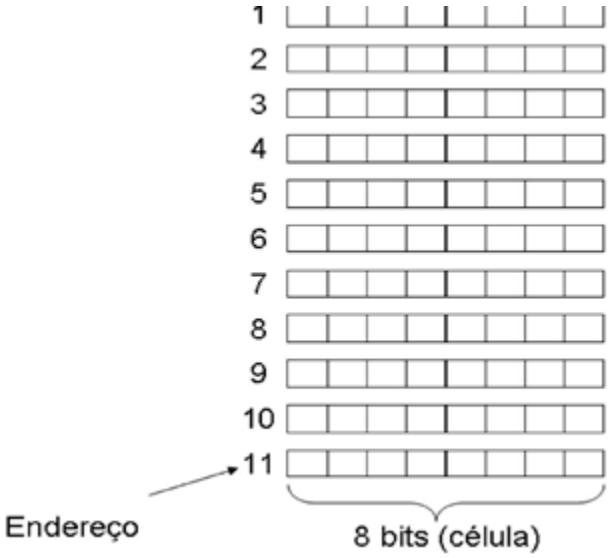


Figura 28

Os processadores operam mais rápidos que as memórias. A função principal do processador é executar instruções e a memória é armazenar valores (dados e instruções para serem executadas), mas ela também precisa operacionalizar algumas funções como busca das informações e envio das mesmas para o processador. Mesmo assim o processador precisa esperar algumas vezes mais de um ciclo de clock pelas informações que estão na memória. A espera se dá pelo processamento da memória para encontrar a informação pedida e pelo barramento por onde esta informação será enviada, pois o barramento também é mais lento que a velocidade de execução do processador.

Ao longo dos anos os fabricantes de memória se preocuparam principalmente em aumentar a capacidade de armazenamento, enquanto que com os processadores os maiores investimentos eram no aumento de velocidade de processamento. Hoje, tanto a velocidade quanto a capacidade são levadas em consideração em novos projetos de memórias principais.

8.1 - MEMÓRIAS CACHE

Os processadores modernos, para aumentar a velocidade de processamento das instruções, utilizam uma memória interna ao chip para diminuir as solicitações a memória principal, consequentemente diminuindo também o acesso ao barramento (falaremos mais de barramento adiante). Estas memórias internas são chamadas de **memórias cache.**

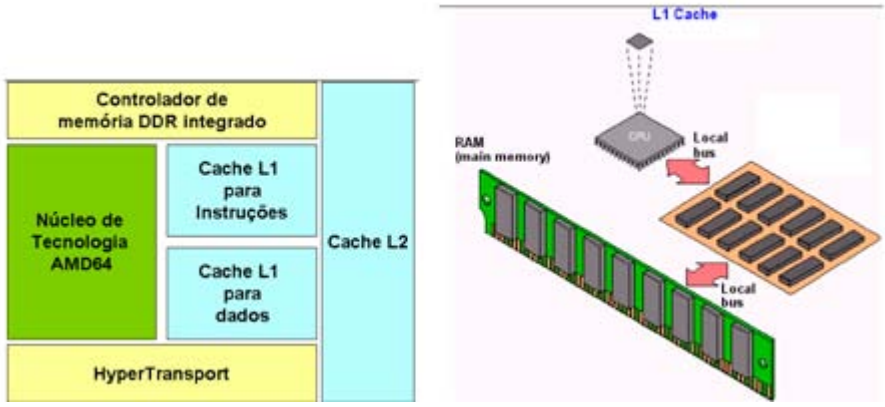
As memórias cache são pequenas memórias, mais rápidas e mais caras, que ficam dentro do chip. Elas não são registradores. Servem para armazenar instruções e dados que podem vir a serem utilizados pelo processador. Somente no caso em que as informações não estejam armazenadas na cache é que a busca se dará na memória principal.

Em algumas experiências em laboratório foi constatado que os programas não acessam a memória de modo totalmente aleatório. O que foi observado é que se um endereço A for referenciado por um programa é muito provável que o próximo acesso seja nas proximidades de A. Tome como exemplo uma sequência de instruções em loop, estrutura muito comum em todos os programas. Estas mesmas instruções serão executadas diversas vezes em um determinado tempo.

Com estas experiências surgiu o conceito do princípio de localidade que consiste em as referências à memória efetuadas durante um intervalo de tempo pequeno tendem a acessar uma parte pequena na memória total. Levando em consideração este conceito, qualquer palavra referenciada por um programa a memória principal é trazida junto com algumas de suas vizinhas. Estas informações são armazenadas na memória cache. A próxima informação solicitada pelo programa executado pode ser uma destas vizinhas. O processador então irá procurar primeiro na cache, mais próxima e rápida, e se não obtiver a informação, solicitará a memória principal. Isto eleva bastante à velocidade de processamento.

A memória cache pode ser classificada também como unificada e dividida. Ela unificada armazena instruções e dados em uma mesma área enquanto que a dividida possui uma área para armazenamento de instruções e outra para os dados. No último caso podemos citar duas vantagens em relação a processadores que utilizam Pipeline. Uma delas é que a unidade de busca de instruções faz acesso às instruções ao mesmo tempo em que a unidade de busca de operandos faz acesso aos dados necessários para a execução de uma outra instrução buscada anteriormente. O modelo de cache unificada não permite acesso paralelo. Além disso, considerando que as instruções não são modificadas durante a execução de um programa, o conteúdo da memória de instruções não precisa ser reescrito na memória principal.

As memórias caches podem ser classificadas também sobre a sua localização no chip. Chama-se cache primária, ou cache L1, a memória que está mais próxima da UAL. Este tipo de cache pode ter tamanho até 512 KB (Pentium 4). A cache secundária, ou cache L2, no início ficava em um chipset na placa-mãe, mas os processadores modernos já estão implantando esta cache dentro do processador, mas fisicamente mais distante da UAL. Ela possui tamanho até 512 KB também, mas é mais lenta que a L1. Existe ainda a cache L3, mais incomum e utilizada no processador AMD K6-III. Este processador possui além da L1 e L2, outra cache na placa- mãe.

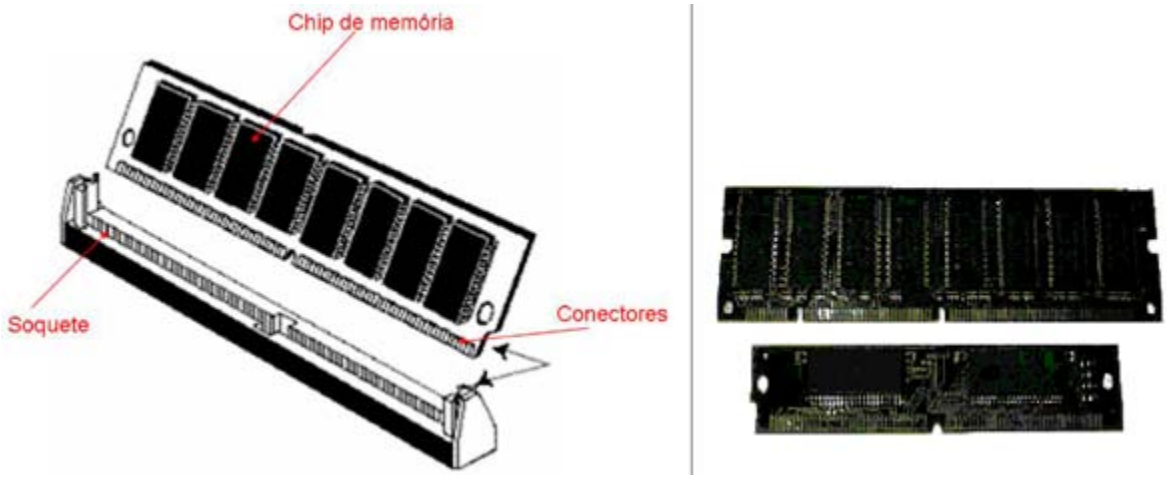


Figuras 29 - Tecnologia AMD64 - Visão geral e 30 - Visão geral de funcionamento cache

8.2 - TIPOS DE ENCAPSULAMENTO DE MEMÓRIA

Até o início dos anos 90, as memórias eram fabricadas em um único chip. Com o passar do tempo o número de chips de uma placa de memória foi aumentando. Os PC`s tinham soquetes vazios onde o usuário poderia plugar mais memória se houvesse necessidade.

Hoje as memórias são placas de circuito impresso com chips montados. O número de chips varia entre 8 e 16 unidades. Estas unidades são chamadas de SIMM (**Single Inline Memory Module**) ou de DIMM (**Dual Inline Memory Module**). A diferença entre estes dois tipos é em relação aos conectores; as memórias SIMM possuem somente conectores em um lado da placa e as memórias DIMM possuem conectores dos dois lados da placa de memória.



Figuras 31 - Esquema de memória e 32 - Memória DIMM e SIMM

Uma configuração SIMM pode ter um total de oito chips de 32 Mb (4 MB) cada. Portanto a memória inteira tem, neste exemplo, 32 MB. As primeiras memórias SIMM possuíam 30 conectores acessando 8 bits de cada vez. Mais tarde apareceram placas SIMM com 72 conectores que acessavam 32 bits de uma vez.

Atualmente as memórias DIMM são o padrão de mercado. As placas DIMM podem acessar até 64 bits de uma vez e podem ter 92 conectores de cada lado, totalizando 184 conectores banhados a ouro. Existem ainda as memórias do tipo SO-DIMM, fisicamente menores, utilizadas em notebooks.

8.3 - SDR, DDR e DDR2

Os três tipos de memória que iremos estudar seguem o padrão DIMM. Para iniciarmos as

diferenças destas memórias precisamos estudar alguns conceitos novos: frequência de operação, barramento de dados e tempo de latência.

A frequência de operação nos dá quantas transferências por segundo um módulo de memória pode executar. Se tomarmos como exemplo uma memória que possui frequência de 100MHz, isso significa que são realizadas 100 milhões de transferências por segundo. A cada transferência de dados precisamos saber quantos bit´s são transferidos. O barramento de dados é que nos informa quantos bit´s são transferidos por ciclo, por exemplo, a memória SDR (**Single Data Rate**) transfere 64 bit´s por ciclo. Por fim a latência que é o número de ciclos que o módulo demora para começar a transferir dados a partir do momento que eles são solicitados e o intervalo a ser aguardado antes de um novo acesso à memória.

As principais diferenças entre as memórias SDRAM, DDR (**Double Data Rate**) e DDR2 são os valores destas taxas, além das diferenças físicas, como podemos observar na tabela abaixo.

	Frequência	Transferência	Latência
SDRAM	66 MHz 100MHz 133 MHz	64 bit´s	2 a 3 tempos
DDR	266MHz 333 MHz 400 MHz	64 bit´s	2 a 3 tempos
DDR2	400 MHz 533 MHz 667 MHz 800 MHz	64 bit´s	3 a 5 tempos

Uma grande diferença entre as memórias SDR e DDR é que a DDR realiza duas transferências de dados por ciclo de clock.

DDR2 é a evolução da tecnologia de memória DDR. A memória DDR2 caracteriza-se por velocidades mais rápidas, larguras de banda de dados mais altas, menor consumo de energia e desempenho térmico aperfeiçoado. Fisicamente a DDR2 é diferente da DDR e os slots não são compatíveis, ou seja, a placa mãe deve possuir slot para DDR2.

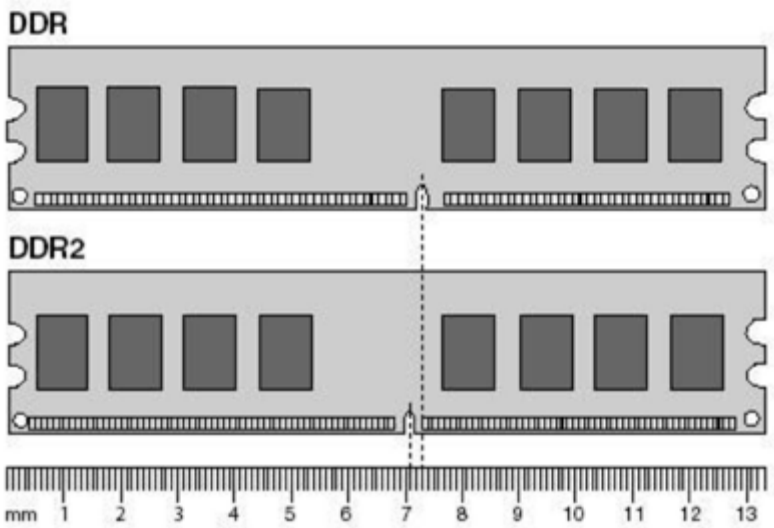


Figura 33 - Diferenças físicas entre DDR e DDR2

Recordando

Aqui vimos uma visão geral da memória principal e seus principais tipos. Os pontos abordados foram:

- Importância da memória principal.
- Memória Cache.
- Tipos de encapsulamento de memória.
- Tipos de memória.

9 - BARRAMENTOS

Um barramento, ou BUS, nada mais é do que um caminho comum pelo qual os dados trafegam dentro do computador. Este caminho é usado para comunicações e pode ser estabelecido entre dois ou mais elementos do computador.

O tamanho de um barramento é importante, pois ele determina quantos dados podem ser transmitidos em uma única vez. Por exemplo, um barramento de 16 bits pode transmitir 16 bits de dado, e um barramento de 32 bits pode transmitir 32 bits de dados a cada vez.

Todo barramento tem uma velocidade medida em MHz. Um barramento rápido permite transmitir dados rapidamente, que tornam as aplicações mais rápidas.

Um PC tem muitos tipos de barramentos, que incluem:

a) Barramento do processador - é o barramento que o chipset usa para enviar/receber informações do processador. O chipset são chips de suporte adjacente contidos na placa-mãe.

b) Barramento de Cache - usado pelos Pentium Pro e Pentium III é um barramento dedicado para acessar o sistema cache. Ele é algumas vezes chamado de barramento Backside. Processadores convencionais que usam placas-mãe de quinta geração tem uma cache conectada ao barramento de memória padrão.

c) Barramento de memória - é um barramento que conecta o subsistema de memória ao chipset e ao processador. Em alguns sistemas o barramento do processador e o barramento de memória são basicamente a mesma coisa.

d) Barramento local de E/S (Entrada/Saída) - é usado para conectar periféricos de alto desempenho à memória, chipset e processador. Por exemplo, placas de vídeo, interface de redes de alta velocidade geralmente usam um barramento deste tipo. Os dois tipos barramentos locais de E/S mais comuns são o VESA Local Bus (VLB) e o Peripheral Component Interconnect Bus (PCI).

e) Barramento padrão de E/S - conecta os três barramentos acima ao antigo barramento padrão de E/S, usado para periféricos lentos (modens, placas de som regulares, interfaces de rede de baixa velocidade) e também para compatibilidade com dispositivos antigos. O principal barramento padrão de E/S é o Industry Standard Architecture (ISA). Nos PC´s, o barramento ISA (antigo padrão) está sendo substituído por barramentos mais rápidos, como PCI. Geralmente, quando se ouve sobre o barramento de E/S, praticamente falamos de slots de expansão.

Alguns novos PCs usam um "barramento" adicional projetado para comunicação de grafismos apenas. A palavra "barramento" foi colocada entre aspas, pois esta tecnologia não se trata de um barramento, ela é um porto: o Accelerated Graphics Port (AGP). A diferença entre um porto e um barramento é que o barramento é geralmente projetado para vários dispositivos compartilhando um meio de comunicação, enquanto um porto é apenas entre dois dispositivos.

Todos os barramentos têm duas partes: um barramento de endereçamento e um barramento de dados. O barramento de dados transfere o dado em si (por exemplo, o valor de memória) e o barramento de endereço transfere a informação de onde o dado se encontra.

9.1 - BARRAMENTO DO PROCESSADOR

O barramento do processador é o caminho de comunicação entre a CPU e o chipset. Este barramento é usado para transferir dados entre a CPU e o barramento principal do sistema. Pelo motivo que o barramento do processador serve para conseguir informações para e da CPU a velocidade mais rápida possível, este barramento opera de forma muito mais rápida que qualquer outro tipo de barramento no computador. Não existem congestionamentos neste barramento. Ele consiste em circuitos elétricos de dados, endereçamento e controle. No processador 80486, o barramento consiste em 32 linhas de endereçamento, 32 linhas de dados e algumas linhas para controle. No Pentium, são 32 linhas de endereçamento, 64 linhas de dados e as linhas de controle.

O barramento do processador opera a mesma velocidade que a CPU opera externamente. Num Pentium 100MHz, o barramento opera a 66,6MHz, bem como num Pentium 133MHz, 166MHz e 200MHz. No processador Pentium II, o barramento de processador opera a 100MHz e no Pentium III, a 100 ou 133 MHz. Ele é ligado nos pinos de conexão externos do processador e pode transferir bits de dados a cada um ou dois ciclos. Um 80486 transfere 32 bits de dados por vez, enquanto um Pentium transfere 64 bits.

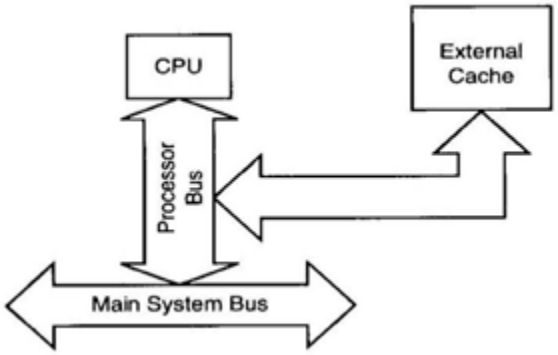


Figura 34 - Funcionamento o barramento do processador

9.2 - BARRAMENTO DE CACHE

Processadores convencionais usam a cache externa na placa-mãe e a CPU acessa esta cache usando o barramento padrão de memória. Para obter um melhor desempenho, a maioria dos novos usam um barramento de alta velocidade conectando o processador à cache L2. No caso do Pentium Pro e Pentium III acima de 600MHz, este barramento é interno ao processador e opera na taxa da CPU. No caso dos Pentium II e III até 600MHz, o barramento de cache opera na metade da taxa do processador. Por exemplo, um Pentium II 266 MHz tem um barramento de cache operando a 133 MHz.

9.3 - BARRAMENTO DA MEMÓRIA

O barramento da memória é usado para transferir informação entre a CPU e a memória principal do sistema (a memória RAM). Estes barramento é uma parte do barramento do processador, ou na maioria dos casos é implementado separadamente por um chipset dedicado que é responsável pela transferência de informação entre o barramento do processador e o barramento da memória.

Devido aos limites da capacidade dos chips de DRAM, a taxa de transferência de informação no barramento da memória é menor que a do barramento do processador.

Vale lembrar que a largura do barramento da memória é a mesma que do barramento do processador. Igualmente, num processador Pentium ou Pentium Pro, que possui um barramento de 64 bit´s, cada banco de memória que é usado deverá ter 64 bits preenchidos. As placas-mãe atuais possuem um suporte a pentes de memória do tipo DIMM, que são

de 64 bit´s. Assim, somente um pente é necessário para o funcionamento do computador.

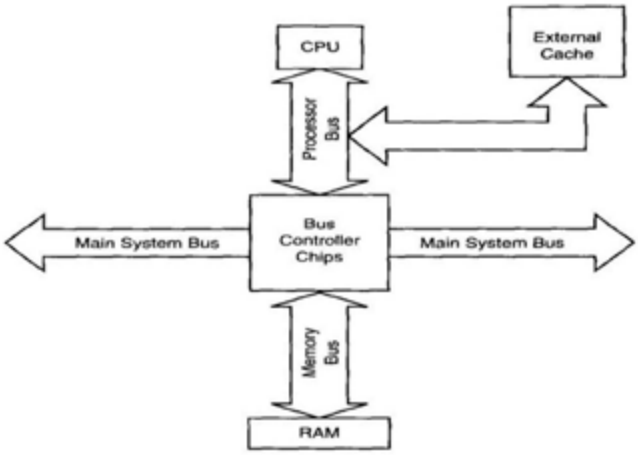


Figura 35 - Funcionamento do barramento da memória

9.4 - BARRAMENTO DE E/S

Desde a criação do PC, muitos barramentos de E/S têm sido desenvolvidos, pela simples razão de CPU´s mais rápidas, maiores demandas de software e maiores requisitos de vídeo. Para cada avanço obtido nos computadores requer-se um barramento que atenda às exigências do hardware. Mesmo assim, necessita-se que seja padronizado diante do mercado a nível mundial, sem grandes custos aos usuários (esta também é uma das razões em que o barramento de E/S não puderam evoluir mais do que se esperava). Até foram criados barramentos de E/S mais rápidos do que o mercado exige, mas não foram colocados com sucesso no mercado justamente pelos custos gerados na atualização de hardware por parte dos usuários.

Pode-se encontrar estes barramento de E/S na arquitetura dos computadores de hoje: ISA, MCA, EISA, VESA Local Bus, PCI Local Bus.

9.5 - BARRAMENTO ISA

Este barramento foi introduzido no PC original em 1981 como um barramento de 8 bit´s e mais tarde foi expandido para 16 bits com o PC/AT em 1984 chamando-se Arquitetura Padrão da Indústria. O padrão ISA é ainda é utilizado em arquiteturas menos recentes, mas este tipo de barramento é geralmente substituído pelo PCI. Mas o padrão ISA ainda é rápido o bastante para muitos periféricos que usamos hoje, como uma placa fax/modem ou uma placa de som.

Existem duas versões do barramento ISA:

- O barramento ISA de 8 bits surgiu com o XT e opera a 4,77MHz. Esta arquitetura de barramento foi usada inicialmente nos primeiros PC´s e é desnecessária nos sistemas atuais. O slot ISA de 8 bits é um conector com 62 contatos possuindo eletronicamente 8 linhas de dados e 20 linhas de endereçamento, habilitando ao slot manusear 1MB de memória (Figura 24).

Figura 36 - Slot ISA de 8 bits

O barramento ISA de 16 bit´s consegue operar num máximo de 8,33MHz, com uma taxa máxima de 8MB/s. Surgiu após a produção de processadores de 16 bits (a partir do 80286, em 1984). Para compatibilidade, a IBM teve que criar um tipo de slot que suportasse ambos 8 bit´s e 16 bit´s. Criou-se um slot com duas divisões. As placas de expansão de 8 bit´s poderiam ser colocadas na parte maior do slot e as de 16 bit´s em ambas as partes do slot. O conector ISA de 16 bits passa a ter, assim, mais 36 pinos de conexão em relação ao ISA de 8 bit´s, em função do amplo caminho de dados. Esse padrão de slots é usado ainda hoje nos computadores Pentium. Todas as placas de CPU possuem três ou quatro slots em cada placa.

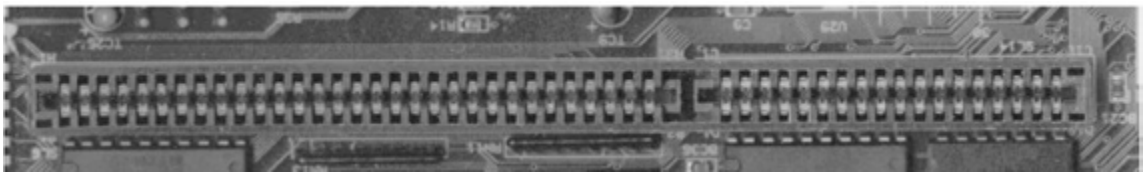


Figura 37 - Slot ISA de 16 bits

9.6 - BARRAMENTO MCA

Com a introdução de chips de 32 bit´s (80386DX) significou que o ISA não poderia suportar todo o poder dessa nova geração de CPU´s. Com o lançamento do 80386DX, a IBM decidiu criar um novo barramento, que resultou no MCA, arquitetura de microcanal. Porém, a IBM não quis somente criar um novo barramento, mas também receber os royalties por ele de outros fabricantes. Outra razão para o MCA não se tornar um padrão da indústria foi que não suportava as placas ISA de 16 bit´s e 8 bit´s, haja vista que o conector era fisicamente menor, mais complexo e bem mais diferente.

9.7 - BARRAMENTO EISA

Este barramento foi criado em 1988 em resposta ao barramento MCA, sendo primeiramente fabricado pela Compaq. Mesmo assim, poucas placas adaptadoras foram criadas a partir do EISA. Simplesmente não emplacou no mercado, assim como a MCA. Porém, o que salvou esse padrão foi o de poder usar as placas ISA presentes. Isto é, os slots EISA estão presentes, mas foram usados principalmente pelas placas ISA, que não necessitaram dos recursos colocados no EISA.

9.8 - LOCAL BUS

Os barramentos de E/S vistos até agora (ISA, MCA e EISA) possuem algo em comum: velocidade baixa. Esta limitação vem dos PC´s originais, onde o barramento de E/S operava a mesma velocidade do barramento do processador. Como a velocidade do processador aumentou o barramento de E/S aumentou apenas em largura da banda de dados, permanecendo a uma velocidade muito mais baixa, por causa da enorme base de placas adaptadoras instaladas que só poderiam operar a velocidades mais baixas.

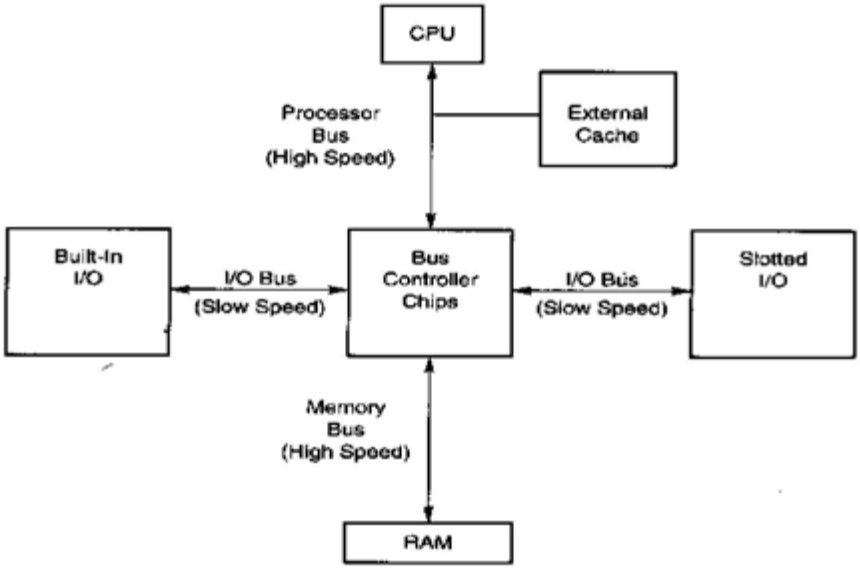


Figura 38 - Layout tradicional do barramento dos PC´s

A ideia de que o sistema rodasse mais lento do que deveria, iria insatisfazer os usuários. Porém, a velocidade do barramento de E/S muitas vezes é só um incômodo, porque não serão necessárias altas velocidades para a comunicação com o teclado ou mouse, por exemplo. O problema só existe na lentidão do barramento quando placas de vídeo e controladores de disco se tornam prejudicados. Com os sistemas e ambientes operacionais gráficos, tanto o vídeo como os discos necessitam de mais velocidade para os dados. Muito mais do que os 8MHz oferecidos com o ISA.

Uma solução óbvia é mover os caminhos de dados dos slots de E/S para a área onde pode acessar velocidades maiores como a do barramento do processador, como um cache externo. Este arranjo torna-se conhecido como local BUS, já que as placas adaptadoras podem acessar diretamente o barramento do processador, que é local. Consequentemente, para comportar esse acesso, os slots necessitam ser diferentes do que os existentes, bem como para diferenciar as placas mais lentas das mais rápidas que são destinadas a esse tipo de barramento.

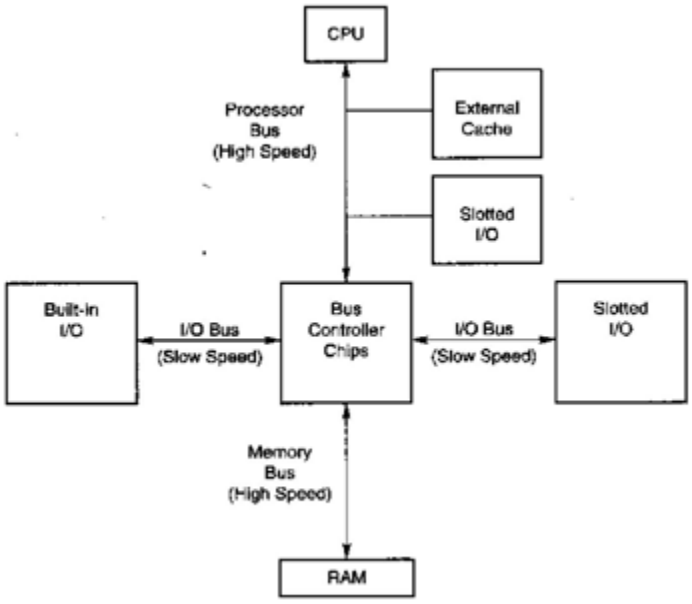


Figura 39 - Funcionamento do barramento local

Nota-se que os primeiros barramento ISA de 8 e 16 bit´s até rodavam a velocidade do barramento do processador, usando este barramento local. Mas quando o processador se tornou mais rápido (e seu barramento também), o barramento principal do ISA teve que ser desacoplado do barramento do processador.

9.9 - VESA LOCAL BUS

VESA Local Bus (VLB) foi o primeiro barramento local popular, em 1992. VESA é abreviação de Video Electronics Standards Association, que foi formada por um grupo de fabricantes interessados em problemas relacionados com vídeos em computadores pessoais. A principal razão do desenvolvimento do VLB foi aumentar o desempenho de vídeo nos PCs.

O VLB é um barramento de 32 bit´s que é de certo modo uma extensão do barramento processador/memória do 486. Um slot VLB é um slot ISA 16 bit´s com um terceiro e quarto slots no final. Desta maneira, uma placa ISA pode ser conectada no VLB.

O VLB roda a taxas de 33 MHz, embora velocidades maiores são possíveis para alguns sistemas. Ele foi muito popular durante a era 486, com a introdução do Pentium e seu barramento local PCI, em 1994, o VLB começou a ser abandonado. Além das vantagens do PCI em relação ao VLB, existem também várias problemas do VLB que contribuíram para seu abandono:

- Seu projeto foi duramente baseado no processador 486, e a adaptação para o Pentium causou sérios problemas de compatibilidade e outros problemas.
- O barramento em si é eletricamente fraco, por exemplo, o número de placas que poderiam ser usadas no barramento era pequeno (normalmente duas ou mesmo uma), e ocorriam problemas quando mais de uma placa era conectada.
- O barramento não tinha um controle muito forte, desde que não havia um bom esquema de árbitro, e não suporta o Plug and Play.

9.10 - BARRAMENTO PCI

PCI é a abreviação de *Peripheral Component Interconnect*, um padrão de barramento local desenvolvido pela Intel Corporation. Embora ele tenha sido desenvolvido pela Intel, o PCI não é ligado a qualquer família de microprocessadores.

A maior parte dos PC´s modernos incluem um barramento PCI em adição ao barramento de expansão ISA. Mas a tendência é que o barramento ISA seja totalmente substituído pelo PCI.

Como o VESA Local Bus, o PCI é um barramento de 32 bit´s que normalmente roda a um máximo de 33 MHz. Nestas condições, ele funciona a 133 MB/s. A vantagem chave do PCI sobre seu predecessor, o VLB é a existência de um circuito no chipset que controla o barramento. Ele era basicamente uma extensão do barramento do processador 486. O PCI deste modo não é casado com 486, e seu chipset fornece funcionalidades para controle, que habilita o PCI a fazer mais coisas que o VLB poderia fazer.

A especificação 2.1 do barramento PCI estende o barramento para 64 bits e opera com a velocidade de 66MHz. Assim, a velocidade pode ser quadruplicada. Na prática o barramento PCI de 64 bit´s ainda não foi implementado no PC (existem em outras plataformas como o Digital Equipment's Alpha). Atualmente a velocidade é limitada a 33 MHz na maior parte dos PC´s.

A operação do barramento PCI pode ser configurada como síncrona ou assíncrona, dependendo do chipset e da placa-mãe. Na configuração sincronizada (usado por maior parte dos PCs), o PCI roda na metade da velocidade do barramento de memória; como este barramento opera usualmente a 50, 60 ou 66 MHz, o PCI poderia rodar a 25, 30 ou

33 MHz, respectivamente. No modo assíncrono, a velocidade do PCI pode ser independente da velocidade do barramento de memória. Isto é controlado por jumpers na placa-mãe, ou pela BIOS.

O barramento PCI opera concorrentemente com o barramento do processador. A CPU pode processar os dados como um cache externo enquanto o barramento PCI está ocupado transferindo informação entre outras partes do sistema.

Além do ser mais eficiente, o barramento PCI incorpora o recurso Plug and Play, não necessitando que o usuário configure as placas adaptadoras. O sistema se encarrega disso. Para isso, os computadores devem possuir um BIOS com essa característica. Por exemplo, o Award Plug and Play BIOS possui esse recurso.

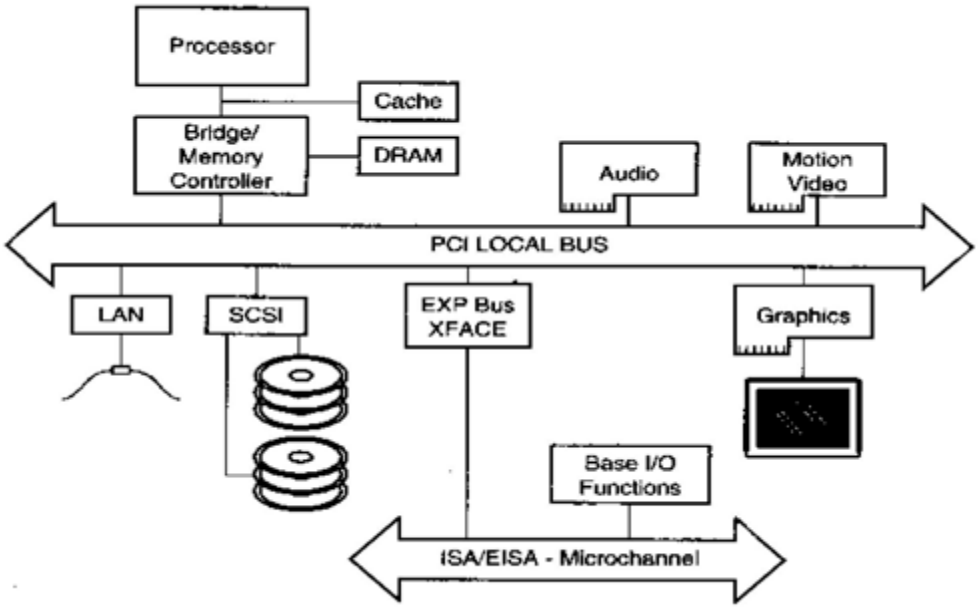


Figura 40 - Estrutura do barramento PCI

Os dados trafegam através do barramento PCI a 33MHz, com toda a banda de dados da CPU, isto é, usando-se uma CPU de 32 bits, a largura é de 32 bits, com uma taxa de transferência de 132MB por segundo no máximo.

Figura 41 - Slot PCI de 32 bit´s

O slot PCI de 64 bit´s é constituído de uma divisão a mais, como nos VLB. Com esta banda de dados, a velocidade dobrando para um máximo de 264MB/s.

9.11 - AGP

A ideia central do AGP é ser um barramento rápido, feito sob medida para o uso das placas 3D de alto desempenho. A versão original do AGP foi finalizada em 1996, desenvolvida com base nas especificações do PCI 2.1. Ela operava a 66 MHz, permitindo uma taxa de transferência teórica de 266 MB/s.

Na época, as placas 3D ainda eram bastante primitivas, de forma que ainda não existia uma demanda tão grande por um barramento mais rápido. Por causa disso, o AGP demorou um

pouco para se popularizar. O primeiro chipset com suporte a ele foi o Intel i440LX lançado no final de 1997, e a adoção ocorreu de forma gradual durante 1998 e 1999.

O padrão AGP inicial não chegou a ser muito usado, pois em 1998 surgiu o padrão AGP 2X, que mantém a frequência de 66 MHz, mas introduz o uso de duas transferências por ciclo (assim como nas memórias DDR) dobrando a taxa de transferência.

Em seguida foi introduzido o AGP 4X e o 8X, que realizam, respectivamente, 4 e 8 transferências por ciclo, atingindo taxas de transferência teóricas de 1066 e 2133 MB/s.
O desempenho de uma placa 3D é fortemente atrelado à velocidade de acesso à memória. Mais de 95% das informações que compõem uma cena 3D de um game atual são texturas e efeitos, que são aplicados sobre os polígonos. As texturas são imagens 2D, de resoluções variadas que são "moldadas" sobre objetos, paredes e outros objetos 3D, de forma a criar um aspecto mais parecido com uma cena real.

A velocidade do barramento AGP é importante quando o processador precisa transferir grandes volumes de texturas e outros tipos de dados para a memória da placa de vídeo, quando a memória da placa se esgota e ela precisa utilizar parte da memória principal como complemento e também no caso de placas de vídeo onboard, que não possuem memória dedicada e, justamente por isso, precisam fazer todo o trabalho usando um trecho reservado da memória principal.

Naturalmente, tudo isso também pode ser feito através do barramento PCI. O problema é que a baixa velocidade faz com que a queda no desempenho seja cada vez maior, conforme cresce o desempenho da placa de vídeo.

Durante muito tempo, fabricantes como a NVidia e a ATI continuaram oferecendo suas placas também em versão PCI, mas a partir de um certo ponto, a diferença de desempenho entre as duas versões passou a ser tamanha que, por mais que ainda existisse uma certa demanda, as placas PCI foram sumariamente descontinuadas.

Outra vantagem do AGP é que o barramento é reservado unicamente à placa de vídeo, enquanto os 133 MB/s do barramento PCI são compartilhados por todas as placas PCI instaladas.

Note que existe uma diferença entre barramento e slot. Uma placa de vídeo onboard é apenas um chip instalado na placa-mãe, ou mesmo um componente integrado diretamente ao chipset e não uma "placa" propriamente dita. Mesmo assim, ela pode ser ligada ao barramento AGP utilizando uma conexão interna. É muito comum ainda que a placa-mãe inclua um chipset de vídeo onboard e, ao mesmo tempo, um slot AGP, que permite instalar uma placa offboard. Neste caso, entretanto, a placa onboard é desativada ao instalar uma placa offboard, já que o AGP não pode ser compartilhado pelas duas placas.

Assim como no caso do barramento PCI, a frequência do barramento AGP está atrelada à frequência de operação da placa-mãe, de forma que, ao fazer overclock aumentando a frequência do FSB, a frequência do barramento AGP sobe na mesma proporção, o que, a partir de um certo ponto pode causar problemas de estabilidade. Entretanto, aumentar a frequência do AGP não tem uma relação direta com o desempenho da placa de vídeo, pois as placas atuais utilizam um circuito de clock próprio e por isso não são influenciadas por mudanças na frequência do barramento. Aumentando a frequência do AGP, melhorando apenas o fluxo de dados entre a placa de vídeo, memória e processador, o que tem pouco efeito nas placas atuais, com slots AGP 4X ou 8X.

Além da questão da velocidade existe também a questão da tensão utilizada. O padrão AGP 1.0 previa placas AGP 1X e 2X, que utilizam tensão de 3.3V. O padrão AGP 2.0, finalizado em 1998, introduziu o AGP 4X e a tensão de 1.5V (utilizada pelas placas atuais), quebrando a compatibilidade com o padrão antigo.

Placas de vídeo que utilizam sinalização de 3.3V possuem o chanfro do encaixe posicionado ao lado esquerdo, enquanto nas placas que utilizam 1.5V, ele é posicionado à direita.
A maioria das placas AGP fabricadas de 2003 em diante são "universais" e podem ser utilizadas tanto nas placas-mãe antigas, com slots de 3.3V, quanto nas placas com slots de 1.5V. Elas possuem os dois chanfros, o que permite que sejam encaixadas em qualquer slot.

A mesma distinção existe no caso das placas-mãe. Placas antigas, que utilizam slots de 3.3V possuem o chanfro à esquerda, enquanto as placas com slots de 1.5V utilizam o chanfro posicionado à direita (em relação aos conectores do painel ATX).
Existem ainda placas com slots AGP universais, em que o slot não possui chanfro algum e por isso permite a instalação de qualquer placa. Nesse caso a placa é capaz de detectar a tensão utilizada pela placa e fornecer a tensão adequada. Elas são mais raras, pois a necessidade de instalar tanto os circuitos reguladores para 1.5V quanto para 3.3V, encarece a produção.

Existe ainda o padrão AGP 3.0 utilizado pelas placas AGP 8X (e parte das 4X), que prevê o uso de sinalização de 0.8V. Nesse caso, entretanto, a transição foi feita de forma transparente, sem que fosse quebrada a compatibilidade com as placas antigas. Ao ser instalado em uma placa-mãe com slot AGP 2.0 (2X ou 4X, 1.5V), a placa de vídeo funcionará normalmente, utilizando sinalização de 1.5V e o modo de transferência mais rápido suportado pela placa-mãe. Caso a placa de vídeo utilize um conector AGP universal, com os dois chanfros, significa que ela pode (pelo menos em teoria), ser instalada mesmo em uma placa-mãe antiga, com um slot de 3.3V.

O termo "em teoria" foi usado, pois existem muitos casos de incompatibilidades entre placas de vídeo AGP de fabricação mais recente e placas-mãe antigas (e vice-versa), mesmo em casos em que os encaixes são compatíveis.

Além dos problemas relacionados às deficiências nos drivers e incompatibilidade por parte do BIOS, existem problemas relacionados à alimentação elétrica, onde a placa de vídeo não indica corretamente qual é a tensão utilizada (fazendo com que a placa-mãe utilize 1.5V para uma placa que trabalhe com 3.3V, por exemplo) ou que a placa-mãe não seja capaz de alimentar a placa de vídeo com energia suficiente. Esse último caso é o mais comum, já que as placas AGP mais recentes consomem muito mais energia que as antigas.
Embora seja mais recente que o PCI e tenha sido largamente utilizado, o AGP é atualmente um barramento em vias de extinção, devido à popularização do PCI-Express. Desde o final de 2006, placas novas com slots AGP são um item raro, com exceção de algumas placas da PC-Chips, ECS e Phitronics.

9.12 - PCI-EXPRESS

O padrão PCI surgiu no início da década de 1990 e, por mais de 10 anos, foi o barramento mais utilizado para a conexão de dispositivos ao computador, principalmente placas de vídeo, placas de som, placas de rede e placas de modem. O barramento PCI trabalha com 32 bit´s por vez na transferência de dados (mas há alguns slots PCI que funcionam a 64 bits), o que permite ao padrão atingir a velocidade (teórica) de até 132 MB por segundo.
Como as aplicações em 3D exigem, cada vez mais, taxas de transferências maiores, o barramento AGP foi inserido no mercado, oferecendo velocidades que vão de 266 MB por segundo (no padrão AGP 1X) à 2128 MB por segundo (no padrão AGP 8X). Praticamente todas as placas-mãe com suporte a AGP só possuem um slot desse tipo, já que este padrão é usado exclusivamente por placas de vídeo.

O problema é que, mesmo oferecendo velocidades acima de 2 GB por segundo, a indústria percebeu que o slot AGP 8X não suportaria os avanços seguintes em aplicações gráficas, não só por limitações na taxa de transferência de dados, mas também pelo uso de recursos que a tecnologia não suporta. Ainda, é necessário considerar que, apesar do AGP ter vantagens bastante razoáveis, seu uso é destinado apenas às aplicações de vídeo. Acontece

que aplicações de áudio e rede, por exemplo, também evoluem.

Na busca de uma solução para esses problemas, a indústria de tecnologia trabalhou (e trabalha) no barramento PCI Express, cujo nome inicial era 3GIO. Trata-se de um padrão que proporciona altas taxas de transferência de dados entre o computador em si e um dispositivo - por exemplo, entre a placa-mãe e uma placa de vídeo 3D.

A tecnologia PCI Express conta com um recurso que permite o uso de uma ou mais conexões seriais, isto é, "caminhos" (também chamados de lanes) para transferência de dados. Se um determinado dispositivo usa um caminho, então diz-se que este utiliza o barramento PCI Express 1X. Se utiliza 4 conexões, sua denominação é PCI Express 4X e assim por diante. Cada lane pode ser bidirecional, ou seja, pode receber e enviar dados.

Cada conexão usada no PCI Express trabalha com 8 bits por vez, sendo 4 bits em cada direção. A frequência usada é de 2,5 GHz, mas esse valor pode variar. Assim sendo, o PCI Express 1X consegue trabalhar com taxas de cerca 250 MB por segundo, um valor bem mais alto que os 132 MB do padrão PCI.

Atualmente, o padrão PCI Express trabalha com até 16X, o equivalente a 4000 MB por segundo. Possivelmente, com o passar do tempo, esse limite aumentará. Já se sabe inclusive que a implementação de um barramento com 32 bits é possível. A tabela abaixo mostra os valores das taxas do PCI Express comparadas às taxas do padrão AGP:

AGP	PCI Express
AGP 1X: 266 MB por segundo.	PCI Express 1X: 250 MB por segundo
AGP 4X: 1064 MB por segundo.	PCI Express 2X: 500 MB por segundo.
AGP 8X: 2128 MB por segundo.	PCI Express 8X: 2000 MB por segundo.
	PCI Express 16X: 4000 MB por segundo.

É importante frisar que o padrão 1X é pouco utilizado e, devido a isso, há empresas que chamam o PCI Express 2X de PCI Express 1X. Assim sendo, o padrão PCI Express 1X pode representar também taxas de transferência de dados de 500 MB por segundo.

9.13 - BARRAMENTO PC-CARD (PCMCIA)

Este barramento é destinado aos computadores portáteis (notebooks e laptops) desenvolvido pela PCMCIA, com um conjunto de mais de 300 fabricantes, onde se estabeleceu os padrões para os cartões adaptadores e de expansão para notebooks e laptops.

Um cartão PCMCIA usa de 68 conectores, sendo ainda mais caro que o ISA. Apesar dos padrões, a indústria flexibilizou demais a arquitetura, de forma que alguns cartões podem não ser compatíveis com algum equipamento ou outro. Os cartões também possuem o recurso de ser Plug and Play.

9.14 - USB

A interface externa USB fornece uma comunicação serial de 12 Mbps, apenas sobre uma conexão de 4 fios. Um único porto USB pode ser usado para conectar até 127 periféricos, tal como mouse, modems, teclados, scanners, câmeras.

A USB também atende às especificações Plug and Play da Intel, inclusive de poder conectar os dispositivos com a máquina ligada e sem precisar reiniciá-las. Simplesmente se conecta o dispositivo e a USB irá detectar automaticamente e alocar os recursos necessários para o seu funcionamento. O suporte a USB é fornecido com o Windows 95 e mais ainda pelo BIOS.

Outra característica importante é que os dispositivos USB conectados têm o fornecimento

de energia pelo próprio USB, o que significa que o USB verifica quais os requisitos de energia requeridos pelos periféricos e avisar se estes dispositivos excederem os limites.

9.15 - FIREWIRE (IEEE 1394)

O FireWire é um barramento serial padrão extremamente rápido que suporta taxas de transferências de dados de até 400 MBps. Um único porto 1394 pode ser usado para conectar até 63 dispositivos externos. Além da alta velocidade, o 1394 também suporta dados isócronos, transmitindo dados com uma taxa garantida. Isto é ideal para dispositivos que necessitam altas taxas de transferências em tempo-real, tal como dispositivos de vídeo.

Embora muito rápido, o 1394 é muito caro. Tal como o USB, o 1394 suporta Plug and Play, e também fornece potências aos periféricos. A principal diferença entre o 1394 e USB é que o 1394 suporta taxas de transferências altas e é mais caro. Por esta razão, é esperado usado mais frequentemente para dispositivos que necessitam taxas altas, como câmeras de vídeo, e o USB.

10 - DISPOSITIVOS DE ENTRADA E SAÍDA

Os dispositivos de entrada/saída são equipamentos utilizados como portadores das informações que o computador irá processar. Por exemplo, quando se pressiona uma tecla, faz com que o teclado transmita o código da tecla pressionada. Este código é recebido por um circuito chamado de INTERFACE DE TECLADO. Ao receber o código de uma tecla, a interface de teclado avisa a CPU que existe um caractere recebido. Por outro lado, quando a CPU precisa enviar uma mensagem para o usuário, precisa que a mensagem seja colocada na tela. Isto é feito com auxílio de um circuito chamado de INTERFACE DE VÍDEO. A CPU envia para a interface de vídeo, a mensagem, seja ela em forma de texto ou figura. A interface de vídeo coloca então a mensagem na tela. Existem, portanto no computador, os chamados DISPOSITIVOS DE ENTRADA E SAÍDA, também chamados de PERIFÉRICOS. Através desses dispositivos, o computador pode armazenar, ler, transmitir e receber dados. Dentre os diversos dispositivos de E/S, existem alguns que são especializados apenas em ENTRADA, outros especializados apenas em SAÍDA e outros em ENTRADA E SAÍDA. Podemos citar os seguintes exemplos:

- **ENTRADA:**

- Teclado - Lê os caracteres digitados pelo usuário.
- Mouse - Lê os movimentos e toque de botões.
- Drive de CD-ROM - Lê dados de discos CD-ROM.
- Microfone - Transmite sons para o computador.
- Scanner - Usado para o computador "ler" figuras ou fotos.

- **SAÍDA:**

- Vídeo - Mostra ao usuário, na tela caracteres e gráficos.
- Impressora - Imprime caracteres e gráficos.
- Alto-falante - Realiza comunicação com o usuário através de som.

- **ENTRADA E SAÍDA:**

- Disco rígido - Grava e lê dados.
- Drive de disquete - Grava e lê dados em disquetes
- Unidade de fita magnética - Grava e lê dados em fitas magnéticas.

Figura 42 - Dispositivos de entrada

Figura 43 - Dispositivos de saída

Figura 44 - Dispositivos de entrada e saída

Os microcomputadores muitas vezes são vendidos com apenas alguns de seus periféricos de uso mais prioritário. Nesse caso, dizemos que se trata de uma CONFIGURAÇÃO BÁSICA. Um exemplo tipo de configuração básica é aquela em que o microcomputador é acompanhado de:

Monitor	Drives	Disco rígido	Teclado	Mouse	Alto-falante interno

O usuário pode, nesse caso, adquirir novos periféricos e realizar sua instalação no microcomputador. Essa tarefa pode ser realizada por usuários mais experientes, ou então por técnicos especializados. Quando instalamos novos periféricos em um computador, dizemos que estamos realizando uma EXPANSÃO. Nesse caso, podem ser instalados, por exemplo:

Impressora	Kit Multimídia (alto-falantes, placa de som e drive de CD-ROM)	Modem	Scanner

11 - PORTAS DE COMUNICAÇÃO

11.1 - PORTAS SERIAIS

A interface serial ou porta serial, também conhecida como RS-232 é uma porta de comunicação utilizada para conectar modems, mouses (ratos), algumas impressoras e outros equipamentos de hardware.

• A IBM ao utilizar o padrão para o seu Projeto do IBM-PC, definiu que apenas 9 pinos seriam necessários (conector DB9). Todavia, manteve-se nos computadores o conector DB25, por ser um padrão da época.

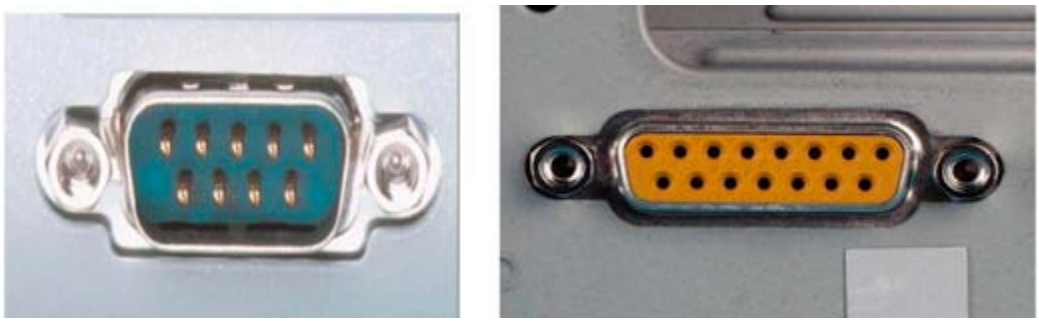


Figura 45

• A porta paralela é uma interface de comunicação entre um computador e um periférico.
• A ideia inicial era conectar uma impressora, mas posteriormente vários periféricos puderam utilizar esta conexão para enviar e receber dados para o computador, um exemplo é o scanner.

Figura 46

11.2 - USB

• USB - Universal Serial Bus (USB) é um tipo de conexão Plug and Play que permite a conexão de periféricos sem a necessidade de desligar o computador. O USB foi concebido na óptica do conceito de Plug and Play, revolucionário na altura da expansão dos computadores pessoais.
• USB 1.0 – 12 Mb/s.
• USB 2.0 – 480 Mb/s.
• USB 3.0 – 4,8 Gb/s (Full-Duplex).

Figura 47

11.3 - FIREWIRE

• O FireWire (também conhecido como i.Link, IEEE 1394 ou High Performance Serial Bus/HPSB) é uma interface serial para computadores pessoais e aparelhos digitais de áudio e vídeo que oferece comunicações de alta velocidade e serviços de dados em tempo real. O FireWire pode ser considerado uma tecnologia sucessora da quase obsoleta interface paralela SCSI.
• Três padrões: 1394, 1394a (FireWire 400), 1394b (FireWire 800).
• FireWire - 400 Mb/s.
• FireWire 800 – 3.2 Gb/s.

Figura 48

12 - REVISÃO

12.1 - SISTEMA BINÁRIO

Como o sistema binário já foi visto faremos somente uma revisão rápida sobre o assunto. O sistema binário foi adotado para a representação de dados em um sistema computacional por representar fielmente como os componentes eletrônicos utilizados em um computador. Todos eles trabalham de forma a estar ligado ou desligado, passando corrente elétrica ou sem nenhuma corrente elétrica, ou seja, se está com valor 1 ou com valor 0.

As memórias utilizadas pelos computadores são um conjunto de componentes eletrônicos que trabalham de forma que estão ligados ou não. Cada componente que pode estar ligado ou não representa a menor unidade de valor do sistema binário, o bit. Os bit's, quando agrupados em pequenos grupos, representam números ou caracteres alphanuméricos (letras). Estes grupos recebem nomes a depender de quantos valores são agrupados:

Chama-se de byte, por exemplo, um conjunto de oito bit's. O byte é a unidade mais conhecida, depois do bit, no sistema binário. A maioria dos cálculos realizados no processador (todos os cálculos são realizados no processador, como você verá mais adiante) utiliza o byte como representação de dados. Mas não é somente o byte que se utiliza em um sistema binário existem mais denominações que também são utilizadas.

O nibble, que é um conjunto de 4 bits, é muito utilizado no sistema hexadecimal. Um dígito hexadecimal precisa exatamente de quatro bits para ser representado. Com um nibble podemos representar até 16 valores distintos. No caso dos números hexadecimais, cada um dos valores 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E e F é representado por quatro bits. Quaisquer 16 valores distintos podem ser representados por um nibble, mas os mais importantes e conhecidos são os dígitos hexadecimais.

Outra denominação utilizada no sistema binário é o word (palavra) que representa um conjunto de 16 bits, ou 2 bytes. O word é muito utilizado quando falamos de valores de memória.

Figura 21- Representação de um Word

O bit 0 é o menos significativo e o bit 15 o mais significativo. Os restantes são referenciados pelos seus números. Observe que o Word é composto por dois bytes. O byte com os bits de 0 a 7 é o byte menos significativo ou de ordem baixa (O.B.) e o byte com os bits de 8 a 15 é o byte mais significativo ou de ordem alta (O.A.).

Com 16 bits é possível obter 216 = 65.536 valores diferentes. Estes podem ser valores numéricos positivos de 0 a 65.535, numéricos com sinal de -32.768 a 32.767 ou qualquer outro tipo de dado que possua até 65.536 valores. Word's são usados principalmente para três tipos de dados: valores inteiros, deslocamentos (offsets) e valores de segmento.

12.2 - OPERAÇÕES LÓGICAS COM BIT´S

Na aritmética binária uma grande parcela dos cálculos é realizada a partir de operações lógicas baseadas em um estudo do matemático George Boole com números binários, por isso as operações lógicas também são chamadas de Álgebra de Boole. Estes tipos de operações trabalham com valores "verdadeiro" e "falso". As principais funções lógicas são AND, OR, XOR e NOT. Veremos um resumo de como funcionam estas operações com bit's.

12.3 - AND (E)

Esta operação determina que uma proposição resultante de duas proposições é verdadeira quando, e só quando, as duas proposições também forem verdadeiras. Vamos tomar como exemplo a frase "se eu estiver cansado **E** tiver um lugar para deitar, então eu durmo". Note que o resultado "eu durmo" só é verdadeiro se "eu estiver cansado" e também se "tiver um lugar para domir". Por exemplo, se eu estiver cansado, mas não tiver um lugar para deitar (negação da segunda proposição) eu não conseguirei dormir.

Vamos representar uma tabela abaixo das operações com as combinações possíveis de valores binários. Tomemos como verdade que o 0 representa falso (não passa corrente elétrica) e que o 1 representa verdadeiro (passando corrente elétrica). A operação AND é diádica, que quer dizer que aceita exatamente dois operandos.

Tabela 1 - Tabela representando a operação AND

AND	0	1
0	0	0
1	0	1

Vamos entender a tabela. Os valores nas células cinza representam os operandos da operação e os que estão nas células brancas são os resultados:

0 AND 0 => 0
0 AND 1 => 0
1 AND 0 => 0
1 AND 1 => 1

Como afirmamos no início, somente temos o resultado verdadeiro (1) se os operandos também sejam verdadeiros.

12.4 - OR (OU)

Ao contrário do AND esta operação determina que uma proposição resultante de duas proposições é falsa quando, e só quando, as duas proposições também forem falsas. Vamos tomar como exemplo a frase "se alguém me falar mal OU riscar meu carro, então eu ficarei muito bravo". Note que o resultado "ficarei muito bravo" é verdadeiro se qualquer uma das proposições iniciais forem verdadeiras. Só teremos um resultado falso (eu não ficarei bravo) se nenhuma das duas proposições iniciais for verdadeira, ou seja, se as duas forem falsas.

Tabela 2 - Tabela representando a operação OR

OR	0	1
0	0	1
1	1	1

12.5 - XOR (OU exclusivo)

Esta operação é um pouco mais complicada para entender. A proposição resultante somente será verdadeira se, e somente se, uma das proposições iniciais forem verdadeiras. Se as duas forem falsas ou se as duas forem verdadeiras, o resultado é falso. Vamos tomar como exemplo a frase "ir a aula de squash XOR ir ao cinema, preciso me decidir". Verifique que não posso ir aos dois lugares ao mesmo tempo (as duas proposições verdadeiras) ou não ir a lugar nenhum (as duas falsas). Eu preciso ir a algum lugar, e tem que ser um dos dois.

Tabela 3 - Tabela representando a operação XOR

XOR	0	1
0	0	1
1	1	0

12.6 - NOT (NÃO)

Esta é a operação mais fácil. Ela nega a proposição inicial, mudando o seu valor. Um não verdadeiro equivale a falso e um não falso equivale a verdadeiro.

- NOT 0 => 1
- NOT 1 => 0

12.7 - OPERAÇÕES LÓGICAS COM BYTES

Os exemplos acima mostraram as funções lógicas funcionando apenas com operandos de bit único. Uma vez que o computador usa grupos de oito, dezesseis ou trinta e dois bits é preciso ampliar a definição destas funções para poder lidar com mais de dois bits. As funções lógicas operam na base do bit a bit, ou seja, tratam os bits da posição 0, depois os bits da posição 1 e assim sucessivamente. É como se fosse uma cadeia de operações. Por exemplo, se quisermos realizar uma operação AND com os números binários 1011 0101 e 1110 1110, faríamos a operação coluna a coluna:

```
          1011 0101
AND       1110 1110
          -----------
          1010 0100
```

Verifique que fizemos a operação AND normal, bit a bit. O processador realiza estas operações a todo o instante. É desta forma que as instruções são executadas em um computador.

12.8 - OPERAÇÕES MATEMÁTICAS COM BITS

As operações matemáticas em bit´s na verdade são operações lógicas como vimos na seção anterior. A soma é representada pela operação OR e a subtração pela operação AND, mas com algumas particularidades.

12.8.1 - SOMA

Para entendermos melhor vamos partir para um exemplo prático:

```
   1100
 +  111
    -----
 =10011
```

Os números binários são base 2, ou seja, somente podem apresentar dois valores: 0 (zero) ou 1 (hum). Na soma de 0 com 1 o total é 1, como no OR. Quando se soma 1 com 1, teoricamente o resultado deveria ser 2, mas como isso não é possível, o resultado é 0 (zero) e passa-se o outro 1 para a "frente", ou seja, para ser somado com o próximo elemento. Vamos fazer um passo a passo da operação descrita acima. Começaremos a operação da direita para esquerda:

0 + 1 => 1

```
0 + 1 => 1
1 + 1 => 0 (passa 1 para frente)
1 + 1 => 0 (passa 1 para frente)
      => 1
```

12.8.2 - SUBTRAÇÃO

Para entendermos, vamos ver uma operação de subtração:

```
   1101110
 -   10111
     -------
 = 1010111
```

Quando temos 0 menos 1, precisamos "emprestar" do elemento vizinho. Esse empréstimo vem valendo 2 (dois), pelo fato de ser um número binário. Então, no caso da coluna 0 - 1 = 1, porque na verdade a operação feita foi 2 - 1 = 1. Esse processo se repete e o elemento que cedeu o "empréstimo" e valia 1 passa a valer 0. Perceba que, logicamente, quando o valor for zero, ele não pode "emprestar" para ninguém, passando-se o "pedido" para o próximo elemento. 0 menos 0 continua sendo 0. 1 menos 1 resulta em 0. 1 menos 0 é igual a 1.

Fique atento quando pedimos "emprestado" do vizinho, pois no "empréstimo" também existe uma operação de subtração. Por exemplo, se for pedir emprestado de 1, então fazemos 1-1=0, como no exemplo anterior. Entretanto, se pedirmos emprestado de 0 temos que faze 0-1, neste caso temos que correr para o outro vizinho, tomando emprestado novamente até obtermos o 1 necessário, sempre realizando uma subtração.

```
   1101100
 -   10111
     -------
 = 1010101
```

12.8.3 - MULTIPLICAÇÃO

A multiplicação entre binários é similar a realizada normalmente, ou seja, multiplicamos cada elemento da base por todos os elemento do valor de cima. A multiplicação é normal: 0 x 0 => 0, 0 x 1 => 0 e 1 x 1 => 1. A única diferença está no momento de somar os termos resultantes da operação, que seguirá a soma binária como visto acima.

Segue um exemplo para entender melhor:

```
       1 0 1 1
      x 1 0 1 0
      ---------
       0 0 0 0
  +   1 0 1 1
  +0 0 0 0
+ 1 0 1 1
  ---------------
= 1 1 0 1 1 1 0
```

12.8.4 - DIVISÃO

Esta operação também é similar à operação com números decimais. Só fique atento as subtrações que seguem o padrão de subtração em binário, como por exemplo:

```
  110 | 10
- 10   11
  --
  010
-  10
   --
   00
```

Sobre a representação de dados podemos falar ainda sobre os inteiros com sinal, onde o bit mais significativo representa se o número é negativo ou positivo, e ainda sobre a representação dos números reais. Este último pode ter duas representações: ponto fixo e ponto flutuante. Atualmente os computadores possuem em seus chips de processamento instruções capazes de realizar operações com pontos flutuantes (os pontos fixos foram substituídos por esta técnica) realizando assim cálculo com números que possuem valores reais.

As operações de ponto flutuante são bastante custosas para o processador por representar números infinitos na mantissa. Entretanto temos um espaço finito para as operações (a memória de um computador não é infinita) por este motivo as operações com ponto flutuante precisam trabalhar constantemente com aproximações, por isso a necessidade de um poder maior de processamento.

Mais informações sobre este tema você encontrará na bibliografia e em outros livros sobre arquitetura de computadores.

Referências

HENESSY, Jonh L. PATTERSON, David A. **Arquitetura de Computadores: Uma Abordagem Quantitativa.** Editora Campus, 2003.

STALLINGS, William. **Arquitetura e Organização de Computadores**. 5ª ed. Editora Prentice- Hall, 2002.

Memória de um computador. Disponível em: http://www.ime.usp.br/~weslley/memoria.htm. Acesso em 1º mar. 2012.

Memórias RAM e ROM. Disponível em: http://www.infowester.com/memoria.php. Acesso em 1º mar. 2012.

Organização de Computadores. Disponível em: http://www.organizacaodecomputadores.kit.net. Acesso em 1º mar. 2012.

alfamacursos.com.br